ICASE

MULTI-GRID SOLUTIONS TO ELLIPTIC FLOW PROBLEMS

Achi Brandt

and

Nathan Dinar

Report Number 79-15

July 16, 1979

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING

NASA Langley Research Center, Hampton, Virginia

Operated by the

UNIVERSITIES SPACE USRA RESEARCH ASSOCIATION

MULTI-GRID SOLUTIONS TO ELLIPTIC FLOW PROBLEMS

Achi Brandt
Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot, Israel

Nathan Dinar
Israel Institute for Biological Research
Ness-Ziona, Israel

ABSTRACT

Various concepts of ellipticity of finite-difference approximations to general elliptic partial differential systems are reviewed and introduced, and rules are given for the construction of stable schemes with high approximation orders, even for singular perturbation problems. Fast multi-grid solvers for these discrete schemes are described. These solvers also provide a convenient way of separating the questions of accuracy and stability (using, for example, both central and upstream differencing). The local mode analysis, which accurately predicts the efficiency of multi-grid solvers, is presented. Concrete examples are given in terms of Cauchy-Riemann equations and the steady-state incompressible Navier-Stokes equations. Their multi-grid solution, based on new "distributive" relaxation schemes, costs about seven work-units.

ACKNOWLEDGMENT

Part of the work reported here was performed under NASA Contract No. NAS1-14101 while the first author was in residence at ICASE, NASA Langley Research Center, Hampton, Virginia 23665. Another part was performed while he visited the Mathematics Research Center, University of Wisconsin, Madison. The second author was supported under Contract No. NAS1-14927-3 while in residence at the College of William and Mary, Williamsburg, Virginia 23185.

TABLE OF CONTENTS

1. INTRODUCTION

The Multi-Level Adaptive Technique (MLAT) is a numerical strategy of solving continuous problems by cycling between coarser and finer levels of discretization. For general partial differential boundary-value problems this technique provides a method for flexible, nearly optimal discretization, together with a very fast solver of the discrete equations. A sequence of uniform grids (or "levels"), with geometrically decreasing mesh-sizes, is employed. The cooperative solution process on these grids involves relaxation sweeps over each of them, coarse-grid-to-fine-grid interpolations of corrections and fine-to-coarse transfers of residuals. MLAT is described in [B3], where historical notes are provided. A more recent brief survey is [B7]. In this article we discuss the construction of discrete approximations to general elliptic boundary-value problems, and their fast multi-grid solutions.

The various multi-grid (multi-level) algorithms used as fast solvers of discrete equations are briefly presented in Section 2.

In Section 3, concepts of ellipticity for discrete systems are extensively discussed. The motivation is the need for general and convenient rules of constructing finite-difference approximations to elliptic systems, with any prescribed order of accuracy, and with sufficient stability. This need is not restricted to multi-grid methods, of course, but the multi-grid algorithm offers new possibilities.

First, in multi-level structures we need to construct difference approximations on *uniform* grids only, since non-uniformity is obtained simply by using non-coextensive levels (see [B5]). We can also assume the boundary to coincide with grid lines (see Sec. 3.5 in [B5]). Furthermore, the questions of accuracy and stability are effectively separated in multi-grid algorithms. Stable approximations are needed in the relaxation phase, while the accuracy is determined by the approximation used in the residual transfers, which itself need not be stable (see Sec. 3.11).

The ellipticity concept for finite-difference equations is more involved than in the differential case, because of its dependence on the mesh-size h: In the Fourier analysis, components with wavelengths smaller than $2h$ are absent, which may exactly be the components that determine the local properties (such as ellipticity) of the differential operator. The common practice of attempting at "positive type" difference approximations is sometimes successful, but lacks generality. Positive type approximations are not available for high-order equations, or for high-order approximations to low-order equations. Furthermore, positive type is neither necessary nor sufficient for stability. The definitions of discrete ellipticity introduced by Thomée [T2], [T3], and Thomée and Westergren [TW] are in a sense too close to the differential definition. They are generally useful only for "sufficiently small" h. For the real mesh-sizes used in many practical problems these definitions do not yield the desired stability properties. We therefore discuss various corrections to these definitions, including the scaled ellipticity concepts mentioned in [B2] and extensively developed by Frank [F1]-[F6]. We then point out that the crucial stability properties depend quantitatively on a certain "measure of ellipticity", which, unlike scaled ellipticity, is useful for singular perturbation problems even when the reduced problem is not elliptic. The study here is made in terms of elliptic *systems* of equations, whereas previous studies of discrete ellipticity treated approximations to just *one* elliptic differential equation.

Readers interested only in the practical aspects of constructing finite-difference systems can skip the first parts of Sec. 3 and go directly to Sec. 3.10, where stable schemes of arbitrary orders are described. In this context high-order "upstream" differencing to singular-perturbation problems is discussed, together with its relation to artificial viscosity. Adding explicit terms of artificial viscosity is another, perhaps preferable, alternative. High-order approximations to equations in divergence form, such as $\nabla\cdot(a(x)\nabla u)$ are also presented. Then, in Sec. 3.11, the multi-grid technique of combining upstream and central differencing is introduced.

The stability of difference operators is intimately connected to their fast multi-grid solution. Specifically, the error can be efficiently smoothed by relaxation only if the difference operator has good "measure of ellipticity". In Sec. 4 we present the theoretical aspects of the multi-grid processes, bringing out that relation between stability and fast multi-grid solutions. The tool being used is the local mode analysis ([B2], [B3]) which predicts the multi-grid performance very precisely, so much so that it is routinely used in optimizing the algorithms and in debugging the programs.

The fine-to-coarse and coarse-to-fine optimal interpolation orders are determined by general simple rules derivable from order-of-magnitude mode analysis, so that in each particular problem all we have to decide is the relaxation scheme. The construction of good relaxation schemes, like the construction of good difference equations, requires expertise and physical insight. But a general computer routine exists which can evaluate the efficiency (i.e., the smoothing rate) of any proposed scheme. The routine, developed in collaboration with Nathan Dinar, is called SMORATE, and is available on [MT].

In the last three sections we describe in detail the discretization and multi-grid solution of three concrete elliptic systems: Cauchy-Riemann equations, steady-state Stokes equations and steady-state incompressible Navier-Stokes equations. This work was also done in collaboration with Nathan Dinar. The description proceeds (as indeed did the research itself) from the simplest system (Cauchy-Riemann) to the more complicated ones. A new type of relaxation, called Distributive Gauss Seidel (DGS), has been developed, which yields a smoothing factor $\bar{\mu} = .5$ for each of these systems. As a result, the multi-grid solution of each system requires about seven work units (i.e., a computational work equivalent to about seven relaxation sweeps over the finest grid). This is true in particular for the incompressible Navier Stokes equations in an arbitrary domain with large Reynolds numbers. The procedure may work even when the steady-state solution is unstable, provided the coarsest grid is fine enough to resolve the unstable modes and the coarsest-grid equations are solved directly.

For the (inhomogeneous) Cauchy-Riemann system there exists an even faster multi-grid solution, which solves the second-order discrete equations to the level of truncation errors in less than 24n additions, where n is the number of discrete unknowns, without using any multiplications or divisions, and without taking advantage of the smoothness of the solution (Sec. 5.5).

Readers familiar with multi-grid procedures can start their reading at Sec. 5, and learn about ellipticity of discrete systems first in terms of the concrete examples.

Multi-grid procedures have been developed for non-elliptic flow problems, such as transonic flows (see [B3], [SB]), and (in a preliminary way only) compressible Navier-Stokes and initial-value problems (see [B7]). For the multi-level adaptive techniques of treating boundary layers and other singularities – see [B5].

2. MULTI-GRID ALGORITHMS

In this chapter we summarize the types of multi-grid algorithms that are currently used as fast solvers for various flow problems. For more explanations the reader is referred to [B3], [B4] and [B5]. We start with the more universally applicable type, the accommodative Full Multi-Grid (FMG) Full Approximation Scheme (FAS) algorithm. The denomination becomes clear later, when we describe algorithms which are not accommodative (fixed algorithms), or not FMG (cycling algorithms), or not FAS (the correction scheme). All the algorithms are based on Cycle C (see [B3], where a sample program is given and explained). Similar algorithms could be based on Cycle A or Cycle B (see Fig. 2 in [B3], and more details in [B2]), but they are not often used.

2.1 Difference equations notation

The differential problem considered is a system of q differential equations

$$L_j \underline{U}(\underline{x}) = F_j(\underline{x}) \ , \quad (\underline{x} \in \Omega \subset \mathbb{R}^d \ , \ j = 1,\ldots,q) \ , \tag{2.1a}$$

and m boundary conditions

$$B_j \underline{U}(\underline{x}) = G_j(\underline{x}) \ , \quad (\underline{x} \in \partial\Omega \ , \ j = 1,\ldots,m) \ . \tag{2.1b}$$

where $\underline{x} = (x_1,\ldots,x_d)$ are the independent variables, $\underline{U} = (U_1,\ldots,U_q)^T$ are the real unknown functions $\underline{F} = (F_1,\ldots,F_q)^T$ and $\underline{G} = (G_1,\ldots,G_m)$ are real known functions (the given data), and L_j and B_j are differential operators, not necessarily linear.

The problem is discretized on a sequence of M levels (grids), with mesh-sizes $h_1, h_2, \ldots, h_M$, where $h_{k+1} = h_k/2$. The discrete approximation to $\underline{U}$ on the k-th level is denoted $\underline{U}^k = (U_1^k,\ldots,U_q^k)^T$. In non-scalar (i.e., $q > 1$) problems, the grids may be "staggered". That is, on the same level k, different functions U_j^k may be defined at different points in each grid cell. (See for example, Figs. 5.1 and 6.1 below.)

The discrete approximation to (2.1) on the k-th level is written in the form

$$L_j^k \underline{U}^k(\underline{x}) = F_j^k(\underline{x}) \ , \quad (\underline{x} \in \Omega_j^k \ , \ j = 1,\ldots,q+m) \ , \qquad (2.2)$$

where the first q equations ($1 \leq j \leq q$) approximate the interior equations (2.1a) and the other m equations approximate the boundary conditions (2.1b). Thus, Ω_j^k is the intersection of the lattice

$$\mathbb{R}_j^{d,k} = \{(a_{j1},\ldots,a_{jd}) + h_k(\nu_1,\ldots,\nu_d) : \nu_i \text{ are integers}\}$$

with Ω (for $1 \leq j \leq q$), or with some extension of $\partial\Omega$ (for $q < j \leq m$). Note that different interior equations may be centered at different grid positions (the lattice shifts $\underline{a}_j$ may be different for different equations j). For purposes of multi-grid processing it is important to keep the discrete equations in a form analogous to the differential equations. This means first that equations (2.2) are written in the difference-quotients form, without, for example, multiplying through by some power of h_k. Such a multiplication can be used in the actual program, but for a correct formulation of the transfer between levels we need the equations in their differential-analog form. Also, we should avoid mixing the boundary conditions and the interior equations.

We will use the vector notation $\underline{L}^k = (L_1^k,\ldots,L_{q+m}^k)^T$, $\underline{F}^k = (F_1^k,\ldots,F_{q+m}^k)^T$, writing (2.2) in the compact form

$$\underline{L}^k \underline{U}^k = \underline{F}^k \ . \qquad (2.3)$$

In the linear case $\underline{L}^k$ can be viewed as a $q \times (q+m)$ matrix of difference-quotient operators independent of $\underline{U}^k$.

Interpolations. The operation of transfer, or interpolation, from level k to level k' will generally be denoted by $I_k^{k'}$. That is, if $\underline{u}^k$ is a function defined on the grid with mesh size h_k, then $I_k^{k'} \underline{u}^k$ is an approximation to $\underline{u}^k$ defined on the grid with mesh size $h_{k'}$. In particular, I_k^{k+1} will denote an interpolation, usually a polynomial (Lagrange) interpolation of some specific order. The order need not be the same for all component functions u_i^k. The order of interpolating u_i^k should not be smaller than the highest order of derivatives of u_i^k in (2.1). When higher-order interpolation is needed (Step B below), we will denote

it by Π. The order of $\Pi_k^{k+1} u_i^k$ should not be less than the largest sum m_i+p_i, where m_i is the order of a derivative of u_i^k and p_i is the approximation order of that derivative. In other words, the Π interpolation should be exact for all polynomials for which the finite-difference approximation is exact. The fine-to-coarse transfers I_k^{k-1} is made by some local averaging; i.e., $I_k^{k-1} u_j^k(\underline{x})$ is some weighted average of values $u_j^k(\underline{y})$ at several points $\underline{y}$ close to $\underline{x}$. These transfers and interpolations are specified in more detail in later chapters, where specific problems are discussed.

The purpose of the multi-grid algorithms described below is to compute a fast approximation to $\underline{U}^M$, the solution to the finest-grid equations. The evolving approximation is denoted $\underline{u}^M$. In the process, equations (2.2) on coarser grids $(k < M)$ will be modified by changing their right-hand side. The modified right-hand sides of the k-th level equations will generally be denoted by $\underline{f}^k = (f_1^k,\ldots,f_{q+m}^k)$ and will depend on $\underline{u}^{k+1}$, the current approximation on the next finer level. The solution to that modified equation will still be denoted by $\underline{U}^k$, and its computed approximation will still be denoted by $\underline{u}^k$.

The meaning of the modified equation and the modified $\underline{U}^k$ depends on the scheme. In the Correction Scheme (CS) $\underline{U}^k$ is designed to be an approximate correction to $\underline{u}^{k+1}$, hence the modified right-hand side will be

$$\underline{f}^k = I_{k+1}^k \underline{r}^{k+1} \tag{2.4}$$

where

$$\underline{r}^{k+1} = \underline{f}^{k+1} - \underline{L}^{k+1} \underline{u}^{k+1} \tag{2.5}$$

is the residual function of the current approximation $\underline{u}^{k+1}$ at the finer level. In the Full-Approximation Scheme (FAS) the designed approximate correction to $\underline{u}^{k+1}$ is $\underline{U}^k - I_{k+1}^k \underline{u}^{k+1}$, hence the modified right-hand side will be

$$\underline{f}^k = \underline{L}^k(I_{k+1}^k \underline{u}^{k+1}) + I_{k+1}^k \underline{r}^{k+1} \quad . \tag{2.6}$$

The difference

$$\underline{\tau}^k = \underline{f}^k - \underline{F}^k \tag{2.7}$$

gives, in this case, an estimate for the local truncation

error on level k ; i.e., an approximation to

$$\underline{\tau}_o^k = \underline{L}^k \underline{U} - \underline{F}^k \quad . \tag{2.8}$$

Since $-\underline{\tau}_o^M$ is the residual of the true differential solution $\underline{U}$ on the finest grid, any computed approximation $\underline{u}^M$ is a satisfactory approximation to $\underline{U}^M$ as soon as the norm of its residuals $\|r^M\|$ is smaller than $\|\underline{\tau}_0^M\|$. Note, however, that $\underline{\tau}^M$ is not computed, since (2.6) and (2.7) hold only when k is not the finest level. But, by Taylor expansions one can get

$$\underline{\tau}_o^k = \underline{a}(x) h_k^p + o(h_k^p) \tag{2.9}$$

where p is the approximation order and $\underline{a}(x)$ is independent of h . Hence

$$\| \underline{\tau}_o^{k+1} \| \approx 2^{-p} \| \tau^k \| \ , \tag{2.10}$$

which can be used to estimate $\| \underline{\tau}_o^M \|$ in our stopping criteria.

2.2 Accommodative FAS Full Multi-Grid (FAS FMG) Algorithm

The Full Multi-Grid (FMG) algorithms, unlike cycling algorithms (see Section 2.6), work themselves up from the coarsest level 1 to the finest M. At each stage we will denote by ℓ the "currently finest" level, that is, the largest k for which an approximate solution u^k has already been computed. By k we will generally denote the current *operation* level. Hence $k \leq \ell$. In "accomodative" algorithms the decision to switch to finer levels or back to coarser ones depends on internal checks, usually based on relative magnitudes of residuals (see Steps E and G below). The steps of the accomodative FAS FMG algorithm, also flowcharted in Fig. 2.1, are as follows.

A. Solving coarsest-grid equations. Set $\ell=1$. Compute an approximate solution u^1 to the coarse grid equations (2.2, $k=1$), either by relaxation or by some direct method. (The term direct method here means a non-iterative solution of linear systems. If the system in nonlinear, the "direct" method will include a few Newton iterations, where the linear system at each iteration is solved directly.)

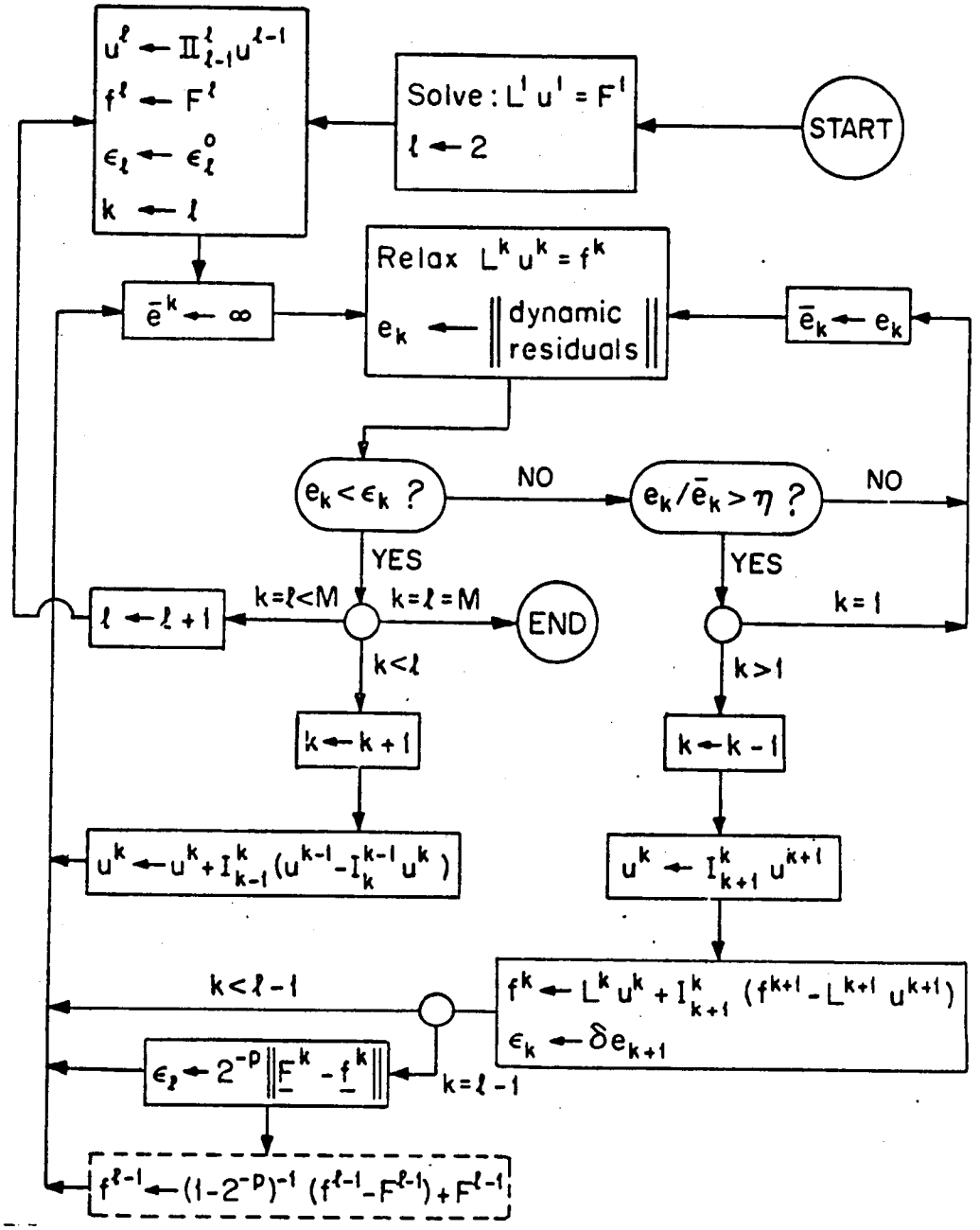


FIGURE 2.1. Accommodative FAS Full Multi-Grid (FAS FMG) Algorithm.

In this version the coarsest-grid (k = 1) correction equations are solved by relaxation. The notation is explained in the text. The operation in the detached-line box (τ-extrapolation) is optional.

B. Setting a new finest level. If $\ell=M$, the algorithm is terminated. If not, increase ℓ by 1. Introduce, as the first approximation for the new finest level, the interpolated function

$$\underline{u}^{\ell} = \Pi^{\ell}_{\ell-1}\underline{u}^{\ell-1} \quad . \tag{2.11}$$

(The higher-order interpolation used here is such that interpolation errors, in any norm, are at most comparable to truncation errors on level ℓ-1.) Set $\varepsilon_{\ell} = \varepsilon^0_{\ell}$, a sufficiently small tolerance. (ε_k serves as the tolerance for solving the k-level equations; see Step E. For ε_{ℓ}, a realistic value is introduced in Step G below, so the value ε^0_{ℓ} is only temporary. $\varepsilon^0_{\ell} = 0$ may be used.) Also set $k = \ell$.

C. Starting a new operation level k. Put $\bar{e}_k = +\infty$ (or a very large number), for reasons to become clear in Step E below.

D. Relaxation sweep. Improve $\underline{u}^k$ by one relaxation sweep. (The role of relaxation is to smooth the error $\underline{U}^k - \underline{u}^k$, so that it can later inexpensively be approximated on a coarser grid. Details of the relaxation sweep and its smoothing power are discussed in later chapters dealing with specific equations. Generally, the sweep may consist of several passes, e.g., one pass for each of the $q+m$ equations (2.2) -- except for those equations which are automatically satisfied, such as Dirichlet boundary conditions.) Concurrently with the sweep, compute some norm e_k of the residuals. (Usually "dynamic" residuals are computed, since they are least expensive: they are almost calculated anyway as part of the relaxation steps.)

E. Testing convergence and its rate. If convergence at the current operation level has been obtained ($e_k \leq \varepsilon_k$), go to Step I. If not, and if the relaxation convergence rate is still satisfactory (i.e., if $e_k \leq \eta\bar{e}_k$, where η is a prescribed factor to be discussed in Sec. 2.3) set $\bar{e}_k = e_k$ and go back to Step D (i.e., continue relaxation. On returning later back to the present Step, $\bar{e}_k$ will contain the previous value of e_k). If however the convergence rate is slow ($e_k > \eta\bar{e}_k$), and the level is not the coarsest ($k > 1$), go to Step F. (The slow convergence implies that the error

$\underline{U}^k - \underline{u}^k$ is smooth, and should therefore be approximated on a coarser level.)

F. Transfer to coarser level. Decrease k by 1. Introduce, as the first approximation for the new (the coarser) level k, the function

$$\underline{u}^k = I_{k+1}^k \underline{u}^{k+1} . \tag{2.12}$$

Define the right-hand side for the new level by

$$\underline{f}^k = \underline{L}^k \underline{u}^k + I_{k+1}^k (\underline{f}^{k+1} - L^{k+1} \underline{u}^{k+1}) , \tag{2.13}$$

which, by (2.12), is the same as (2.6). As the tolerance for this new problem, set $\varepsilon_k = \delta e_{k+1}$. (Since the coarse-grid solution is designed to correct the fine grid solution, its residuals should be smaller than those at the fine grid, but there is no point in having them much smaller. $\delta=0.2$ is usually small enough. See also Sec. 4.7.2.)

G. Finest level stopping parameters. Concurrently with the computation of $\underline{f}^k$, calculate also the norm of $\| \underline{f}^k - \underline{F}^k \|$, using the same norm as used for the dynamic residuals (see Step D). If $k=\ell-1$, set

$$\varepsilon_\ell = 2^{-p} \| \underline{f}^k - \underline{F}^k \| , \tag{2.14}$$

where p is the order of approximation (cf. (2.10). We will thus stop on level ℓ when its residuals become comparable to the truncation errors).

H. Coarse-level solution. If $k=1$ (the coarsest level), one may like to solve the problem directly (see Step A above) and go to Step I. Otherwise, go to Step C.

I. Employing a converged solution to correct a finer level. If $k=\ell$, go to Step B. If $k<\ell$, make the correction

$$\underline{u}_{NEW}^{k+1} = \underline{u}_{OLD}^{k+1} + I_k^{k+1} (\underline{u}^k - I_{k+1}^k \underline{u}_{OLD}^{k+1}) \tag{2.15}$$

where I_{k+1}^k must be identically the same operator as in (2.12). Then increase k by 1 and go to Step C.

2.3 Switching and Stopping Criteria.

A good general value for the fine-to-coarse switching parameter η (see Step E) is

$$\eta = \max_{\underline{x}} \bar{\mu}(\underline{x}) \tag{2.16}$$

where $\bar{\mu}$ is the smoothing factor per sweep (see Sec. 4.1

below), which may vary over the domain if the problem is nonlinear or if its coefficients are not constant. Actually, the precise value of η (as well as that of δ, in Step F) is not critical. If coarse-grid corrections are not efficient enough, η may always be increased a little, a safe value being for example $\eta = \bar{\mu}^{1/2}$. If $\bar{\mu}$ has large variations over the domain, the switching test $(e_k \leq \eta \bar{e}_k)$ can be made separately in subdomains, possibly resulting in *partial relaxation sweeps*, i.e., sweeps which are confined to some subdomains. (Partial sweeps may be important only in multi-grid processes, and not in pure relaxation processes, because it is only with respect to smoothing, and not with respect to convergence, that errors at separate subdomains are practically decoupled from each other.)

In non-scalar $(q>1)$ problems, the different relaxation passes for different interior equations j $(1 \leq j \leq q$ in (2.2)) may have different smoothing factors, in which case one can make more passes on some of them, provided there is no serious feedback of errors from those equations to others. (See an example in Sec. 7.3.) Similarly, even in scalar $(q = 1)$ problems, relaxation passes over boundary equations $(q < j \leq q+m$ in (2.2)) may have slower smoothing than the interior passes. In such a case one can make several (usually two) boundary passes for each interior pass.

Generally when we can ignore coupling (or feedback) between the smoothing processes of the different equations, then for every N_α passes on equation α there should be made N_β passes on equation β, where $N_\alpha \log \bar{\mu}_\alpha$ roughly equals $N_\beta \log \bar{\mu}_\beta$, $\bar{\mu}_j$ being the smoothing factor of equation j. Where the relaxation passes are coupled, however, the optimal strategy should in principle be based on minimizing the overall smoothing factor per work unit; i.e., minimizing

$$\bar{\mu} = \max_{\underline{x}} \max_{\frac{\pi}{2} \leq |\underline{\theta}| \leq \pi} |\mu(\underline{\theta}, \underline{x})|^{1/N} \qquad (2.16a)$$

where $\mu(\underline{\theta},\underline{x})$ is the largest eigenvalue of the amplification matrix (see Sec. 4.1) of the compound sweep (including possibly several passes on some equations), and N is the number of work-units in that sweep. In practice, however, only small values of N_α, N_β and N should be used so as not to approach the limits of the smoothing process (see Sec. 4.3).

2.4 Truncation Extrapolation.

At convergence, it is easy to see that the approximate local truncation error (see (2.7)) is

$$\underline{\tau}^k = \underline{L}^k I_k^\ell \underline{U}^\ell - \underline{F}^k . \tag{2.17}$$

Thus, $-\underline{\tau}^k$ is the residual of the "currently finest" solution $\underline{U}^\ell$ in the level-k equations. It follows from (2.9) that

$$\begin{aligned} \underline{\tau}^k &\approx \underline{\tau}_o^k - \underline{\tau}_o^\ell \\ &\approx a(h_k^p - h_\ell^p) \\ &\approx (1-(h_\ell/h_k)^p)\underline{\tau}_o^k , \end{aligned} \tag{2.18}$$

where $\approx$ indicates equality up to higher-order terms in h_k and h_ℓ. Up to such higher order terms, we can therefore replace $\underline{\tau}^k$ by the more accurate $\underline{\tau}_o^k$. The place to do it is with Step G: If $k=\ell-1$ we replace the right-hand side of the new level by

$$\hat{f}_j^{\ell-1} = (1-2^{-p_j})^{-1} (f_j^{\ell-1} - F_j^{\ell-1}) + F_j^{\ell-1} , \quad (1\le j\le q+m) \tag{2.19}$$

where p_j is the approximation order of the j-th equation. For $k<\ell-1$, (2.19) should not be used again, i.e., using (2.13) in the usual way would now give

$$\underline{\hat{f}}^k = \underline{L}^k \underline{u}^k + I_{k+1}^k(\underline{\hat{f}}^{k+1} - L^{k+1} \underline{u}^{k+1}) , \quad (k<\ell-1), \tag{2.20}$$

so that the correction in $\underline{f}^{k+1}$ (approximately equal to $\underline{\tau}_o^\ell$) is automatically transferred to the coarser levels.

Replacing $\underline{f}$ by $\underline{\hat{f}}$ is called "local truncation extrapolation", or briefly τ-extrapolation. It is a trivial addition to the algorithm, and can improve the solution very much, even in cases where global-truncation (Richardson) extrapolation cannot, since *global* error expansions in terms of h do not always exist.

Note that for staggered grids the interpolation I_{k+1}^k in

(2.12) is not trivial. For τ-extrapolation it should be of the suitable higher-order. For further discussion and results of τ-extrapolations, see [B5] and [D2].

2.5 Fixed Algorithms

A "fixed", in contrast to "accomodative", algorithm is one in which the internal checks (in Step E above) are replaced by pre-assigned flow: the switch to the coarser level $k-1$ is made when a pre-assigned number N_c of relaxation sweeps on level k has been completed. The parameter N_c, like η, depends on the smoothing factor $\bar{\mu}$. A good choice seems to be $N_c=[\log.1/\log\bar{\mu}]$. Similarly, the switch to a finer level $k+1$ may be made as soon as a total of N_F relaxation sweeps has been made on level k since the last "visit" to the finer level. (Thus, if $\nu N_c \le N_F < (\nu+1)N_c$, then ν switches from k to the coarser level $k-1$ are performed before switching to $k+1$.) In some cases $N_F=N_F^k$ depends on the level k.

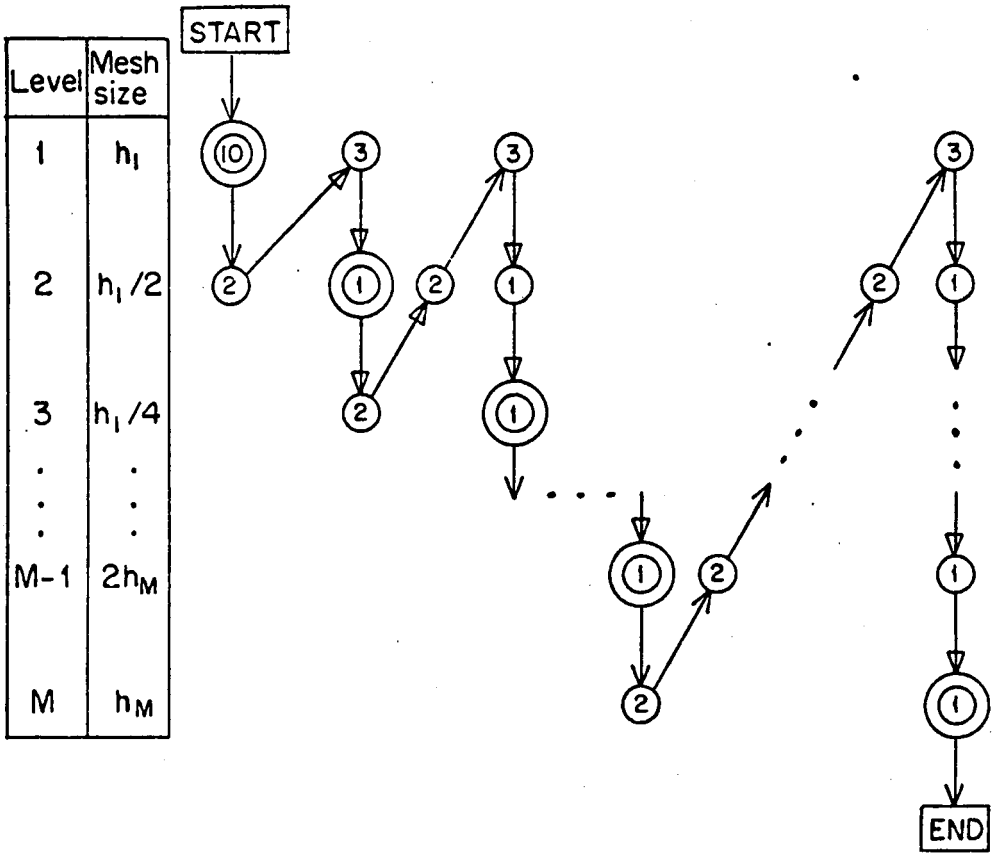
An example of the flow of a fixed algorithm is shown in Fig. 2.2. This flow is suitable for solving any problem with a relaxation smoothing factor $\bar{\mu} \le 2^{-p/d}$ (as, for example, in the standard case of Poisson equation and in practically all the problems discussed in this paper). This algorithm performs a total of $3(M+1-k)$ sweeps on level k, $k=1,\ldots,M$.

The main advantage of fixed algorithms is in saving the work of computing the dynamic residuals at each relaxation sweep. This is a signficant saving when the problem is very simple. For example, for 5-point Poisson problems about 40% of the relaxation time is saved. In more complicated problems, like the Navier-Stokes equations, the saving is quite marginal.

In simple problems, where the fixed algorithms are needed, they are as efficient (per sweep) as the "accomodative" ones. In fact, the latter behave like fixed.

2.6 The Correction Scheme (CS) and Cycling Algorithms.

For linear problems, instead of the Full Approximation Scheme (FAS) used above, we can use the Correction Scheme, in which, for $k<\ell$, $I_k^{k+1}u^k$ is designed to be the correction to u^{k+1}. The only changes to the algorithm are as follows.



= Higher order interpolation (2.11)

= Linear interpolation of corrections (2.15)

= Transfer to coarser level (2.12) - (2.13)

= Transfer to coarser level (2.12) - (2.13) where τ-extrapolation (2.19) can be made

(r) = r relaxation sweeps

= The stage in the algorithm where the error is smaller than the discretization error of that level

FIGURE 2.2. Fixed Full Multi-Grid Algorithm.

In Step F, (2.12) is replaced by $\underline{u}^k=0$, and (2.13) by $\underline{f}^k=I_{k+1}^k(\underline{f}^{k+1}-\underline{L}^{k+1}\underline{u}^{k+1})$. In Step I replace (2.15) by $u_{NEW}^{k+1}=u_{OLD}^{k+1}+I_k^{k+1}u^k$. The tolerance ε_ℓ computed in Step G should either require a special calculation by

$$\varepsilon_\ell = 2^{-p} \| \underline{L}^k I_{k+1}^k \underline{u}^{k+1} - I_{k+1}^k \underline{L}^{k+1} \underline{u}^{k+1} \| ; \qquad (2.21)$$

or be prescribed by

$$\varepsilon_\ell = 2^{-p} \overset{\circ}{e}_\ell , \qquad (2.22)$$

where $\overset{\circ}{e}_\ell$ is the value of the (dynamic) residual norm at the very first relaxation on level ℓ . Or else, arbitrarily small ε_ℓ may be prescribed for cycling purposes.

"Cycling", in contrast to Full Multi-Grid (FMG) algorithms, are simplified algorithms which start when a first approximation $\underline{u}$ is already given on the finest grid. It may be a trivial approximation, such as $\underline{u}^{\circ M}=0$. Or it may be the approximate solution of some other, related, "previous" problem (as in continuation processes, or in evolution problems, or in optimization procedures, etc.). Thus, in cycling algorithms one sets $k=\ell=M$, $\varepsilon_M=\varepsilon_M^0$ and starts at Step C of the above algorithm.

Correction-Scheme and cycling algorithms are usually used in the first stages of developing a multi-grid code, because they are simpler to write and debug. In particular, a cycling algorithm which is allowed to run more cycles than usually needed (ε_M^0 is small, and Step G is skipped) is very useful since it shows the asymptotic multi-grid convergence factor. This factor can be independently calculated by the local mode analysis (see Secs. 4.6, 4.7.5) and is therefore very basic in detecting conceptual errors as well as bugs in the program.

A general easy way for converting CS programs to FAS is described in [B6], Lecture 12.

3. ELLIPTIC DIFFERENCE EQUATIONS AND SYSTEMS

The boundary-value problems discussed in this paper are elliptic. In this chapter various concepts of ellipticity and the construction of elliptic finite-difference systems are discussed. Some of these concepts are important in understanding smoothing properties of relaxation schemes. Readers interested only in the practical aspects of constructing difference equations are referred directly to Secs. 3.10 and 3.11.

3.1 Ellipticity of the differential system

We first review the concept of ellipticity for a system of differential equations. This will be defined for linear systems. Nonlinear systems of equations will be called elliptic if the linearized equations are elliptic. We consider real systems only: complex systems can be rewritten as real ones.

In case the differential system (2.1a) is linear, we can write it in the form

$$\sum_{\beta=1}^{q} \ell_{\alpha\beta}(\underline{x},\underline{\partial})U_\beta(\underline{x}) = F_\alpha(\underline{x}) \quad , \quad (\underline{x} \in \Omega \subseteq \mathbb{R}^d;\ \alpha = 1,\ldots,q), \tag{3.1}$$

$$\ell_{\alpha\beta}(\underline{x},\underline{\partial}) = \sum_{\underline{\gamma}} a_{\alpha\beta\underline{\gamma}}(\underline{x})\underline{\partial}^{\underline{\gamma}} \quad , \tag{3.1a}$$

where $\underline{x} = (x_1,\ldots,x_d)$, $\underline{\partial} = (\partial_1,\ldots,\partial_d)$, $\underline{\gamma} = (\gamma_1,\ldots,\gamma_d)$, $\partial_j = \partial/\partial x_j$, γ_j is a nonnegative integer, $(j = 1,\ldots,d)$, and

$$\underline{\partial}^{\underline{\gamma}} = \partial_1^{\gamma_1}\,\partial_2^{\gamma_2}\ldots\partial_d^{\gamma_d} \quad .$$

We assume $q \geq 1$ and $d \geq 2$.

Roughly speaking, a system is to be called elliptic at a point $\underline{x}$ if the differential equations at $\underline{x}$ are such that boundary value problems with these equations are uniquely and stably solvable, at least in small enough rectangular domains; or, equivalently, the system (3.1) is to be called elliptic if it is solvable for all Fourier components $F_\alpha(x) = A_\alpha \exp(i\,\Xi\cdot\underline{x})$, at least for sufficiently high $\|\Xi\|$, where

$$\Xi\cdot\underline{x} = \Xi_1 x_1 + \ldots + \Xi_d x_d \ , \quad \|\underline{\Xi}\| = (\Xi_1^2 + \ldots + \Xi_d^2)^{\frac{1}{2}} \ ,$$

and $\alpha = 1,\ldots,q$. Thus, the ellipticity of (3.1) depends on the regularity for sufficiently high $\|\underline{\Xi}\|$, of the *characteristic matrix*

$$\ell(\underline{x},\underline{\Xi}) = \{\ell_{\alpha\beta}(\underline{x},i\underline{\Xi})\}_{\alpha,\beta=1,\ldots,q} \quad , \tag{3.2}$$

which is the Fourier transform of (3.1). The *characteristic form* (or *symbol*) and the *principal characteristic form* (or *principal symbol*) of system (3.1) are defined, respectively, as

$$\tilde{L}(\underline{x},\underline{\Xi}) = \det \ell(\underline{x},\underline{\Xi}) \tag{3.3}$$

and

$$\hat{L}(\underline{x},\underline{\Xi}) = \det'\ell(\underline{x},\underline{\Xi}) \quad , \tag{3.4}$$

where det' is the principal part of the determinant defined as follows: If m' is the maximal degree (in $\underline{\Xi}$) of polynomials of the form $\ell_{1i_1}\ell_{2i_2}\cdots\ell_{qi_q}$, where $(i_1,\ldots,i_q)$ is a permutation of $(1,\ldots,q)$, then det' is the part of the determinant of degree m' . Thus, $\hat{L}$ is the dominating part in $\tilde{L}$ as $\|\Xi\| \to \infty$. Hence:

The system (3.1) is called elliptic at the point $\underline{x}$ *if* $\hat{L}(\underline{x},\underline{\Xi}) \neq 0$ for all real $\underline{\Xi} \neq 0$. (3.5)

In this generality, the definition is due to Douglis and Nirenberg [DN].

The condition (3.5) immediately implies that $m' = 2m$ is an even positive integer (the trivial case $m' = 0$ being excluded), and

$$0 < c(\underline{x}) \|\Xi\|^{2m} \leq |\hat{L}(\underline{x},\underline{\Xi})| \leq C'(\underline{x}) \|\Xi\|^{2m} \quad . \tag{3.5a}$$

The system (3.1) is called *elliptic of order* 2m in Ω , if (3.5) holds, with the same $m' = 2m$, at all points $\underline{x} \in \Omega$. It is called *uniformly elliptic of order* 2m if (3.5a) holds uniformly, i.e., if there exist positive constants C and C' such that

$$C \|\Xi\|^{2m} \leq |\hat{L}(\underline{x},\underline{\Xi})| \leq C' \|\Xi\|^{2m}, \quad \text{for all real } \underline{\Xi} \neq 0 \text{ and } \underline{x} \in \Omega \ . \tag{3.6}$$

Elliptic systems of order 2m require m boundary conditions all around the boundary, as can easily be seen by a Fourier analysis of the homogeneous system (having the prin-

cipal part only) with constant coefficients near a hyperplane boundary. Such an analysis also easily shows which sets of m conditions are suitable. Such a set is called "complementary" to the differential equations (see [ADN]). It can be proved that any linear uniformly-elliptic system with m complementary boundary conditions (i.e., with conditions that are complementary at each boundary point to the principal part of the differential system at that point) can be solved on any sufficiently small domain, while on general domains such a problem has a discrete spectrum (discrete sequence of eigenvalues) and all the smoothness properties one can expect. Namely, if the coefficients in the equations and in the boundary conditions, as well as the boundary itself, are all sufficiently smooth, one can prove various a priori estimates for derivatives of $\underline{U}$, which are as strong as one can expect. For example, the following Schauder-type estimate is proved in [ADN] :

$$|U_\beta|^{\Omega}_{j+t_\beta+\delta} \leq$$

$$\leq K\Big(\sum_{\alpha=1}^{q} |F_\alpha|^{\Omega}_{j-s_\alpha+\delta} + \sum_{k=1}^{m} |G_k|^{\partial\Omega}_{j-r_k+\delta} + \sum_{\alpha=1}^{q} |U_\alpha|^{\Omega}_{0} \Big), \quad (3.7)$$

$$(\beta = 1,\ldots,q) ,$$

where K is independent of $\underline{U}$, $\underline{F}$ and $\underline{G}$. Here $t_\alpha \geq 0$ and $s_\beta \leq 0$ are integers such that the degree (in $\underline{\Xi}$) of $\ell'_{\alpha\beta}(\underline{x},i\underline{\Xi})$ is $s_\alpha+t_\beta$, where $\ell'_{\alpha\beta}$ contains those terms in $\ell_{\alpha\beta}$ that enter the principal part. r_k are integers such that the order of differentiating U_α in the k-th boundary condition is at most r_k+t_α. For any domain D, and any integer J and constant $0 < \delta < 1$, the norm $|\varphi|^{D}_{J+\delta}$ of a function φ in the domain D is defined by

$$|\varphi|^{D}_{J+\delta} = \sum_{j=0}^{J} \sup_{\substack{\underline{x}\in D \\ |\underline{Y}|=j}} |\partial^{\underline{Y}} \varphi(\underline{x})| + \sup_{\substack{\underline{x},\underline{y}\in D \\ |\underline{Y}|=J}} \frac{|\partial^{\underline{Y}} \varphi(\underline{x}) - \partial^{\underline{Y}} \varphi(\underline{y})|}{|\underline{x}-\underline{y}|^{\delta}} .$$

We see in (3.7) that $\underline{U}$ is only as smooth as logically allowed by the boundary conditions G. This is of course expected. But solutions to elliptic systems have also the distinct property that their smoothness in the interior of Ω, away from the boundary, does not really depend on the smoothness of the boundary and the boundary conditions. For example,

the following Schauder-type *interior* estimate can be proved:

$$|U_\beta|^{\Omega'}_{j+t_\beta+\delta} \le K' \left(\sum_{\alpha=1}^{q} |F_\alpha|^{\Omega}_{j-s_\alpha+\delta} + \sum_{\alpha=1}^{q} |U_\alpha|^{\Omega}_{0} \right) , \qquad (3.7a)$$

where Ω' is any compact subset of Ω, and K' depends on the distance from $\partial\Omega'$ to $\partial\Omega$. (For the precise nature of this dependence, see [DN].)

The factors K and K' in (3.7)-(3.7a), and in similar stability and smoothness estimates, do not depend on $\underline{F}$, $\underline{G}$ and $\underline{U}$, but they do obviously depend on d, q, m, s_α and t_β, and on the size and smoothness of the coefficients. K also depends on r_k, the size and smoothness of the coefficients in the boundary conditions, and on the smoothness of the boundary itself. Except for those, the sizes of K and K' depend only on some measure of the ellipticity of the system. We will need such a measure in judging quantitatively the fitness of difference schemes, so, for the differential case, we introduce such a measure here.

The largest value of C/C' for which (3.6) holds may be used as a measure for the ellipticity of the system. It can also be defined directly in terms of the (full) characteristic form $\tilde{L}(\underline{x},\underline{\Xi})$, a definition which will be more useful in the discrete case. Thus, we call

$$E(L,\underline{x}) = \lim_{Q\to\infty} \min_{\|\underline{\Xi}'\|,\ \|\underline{\Xi}\|=Q} \left| \frac{\tilde{L}(\underline{x},\underline{\Xi})}{\tilde{L}(\underline{x},\underline{\Xi}')} \right| \qquad (3.8)$$

the *ellipticity measure* of the system (3.1) at the point $\underline{x}$. $E(L) = \min_\Omega E(L,\underline{x})$ is called the ellipticity measure of (3.1) in the domain Ω. When the principal part of L is a power of Laplace operator, $E(L)$ attains its maximal value $E(L) = 1$. When $E(L)$ decreases to 0, the operator approaches local instability: Some small high-frequency perturbations (in $\underline{F}$) are magnified (in $\underline{U}$) much more than other high-frequency perturbations. An example of such an operator is

$$L = -\varepsilon \frac{\partial^2}{\partial x_1^2} - \frac{\partial^2}{\partial x_2^2} - \ldots - \frac{\partial^2}{\partial x_d^2} , \quad (0 < \varepsilon \ll 1) \qquad (3.9a)$$

for which $E(L) = \varepsilon$. Operators with such a degeneracy in

their principal part can be called *degenerate elliptic*. They should be distinguished from *singular perturbation operators* in which the small parameter (ε) multiplies the entire principal part. Typical singular perturbation elliptic operators are the scalar operators

$$L = -\varepsilon\Delta + \underline{a}\cdot\underline{\partial}$$

$$= -\varepsilon\left(\frac{\partial^2}{\partial x_1^2} + \ldots + \frac{\partial^2}{\partial x_d^2}\right) + a_1\frac{\partial}{\partial x_1} + \ldots + a_d\frac{\partial}{\partial x_d} \qquad (3.9b)$$

$$L = -\varepsilon\Delta + w^2 \qquad (3.9c)$$

and

$$L = -\varepsilon\Delta - w^2 \ . \qquad (3.9d)$$

3.2 Finite-difference operators and symbols

On a uniform grid with meshsize h we introduce the translation operators $\underline{T}_h = (T_{h1},\ldots,T_{hd})$, defined by

$$\underline{T}_h^{\underline{\nu}}\, u(\underline{x}) = T_{h1}^{\nu_1} \ldots T_{hd}^{\nu_d}\, u(\underline{x}) = u(\underline{x}+\underline{\nu}h) \ ,$$

where $\underline{\nu} = (\nu_1,\ldots,\nu_d)$ will usually have integer or half-integer components. In terms of T_h we can define the finite difference operators

$$\delta_j^h = T_{hj}^{\frac{1}{2}} - T_{hj}^{-\frac{1}{2}}$$

$$\delta_{jj}^h = (\delta_j^h)^2 = T_{hj} - 2 + T_{hj}^{-1} \ ,$$

the averaging operator

$$\mu_j^h = (T_{hj}^{\frac{1}{2}} + T_{hj}^{-\frac{1}{2}})/2$$

and the difference-quotients operators

$$\partial_j^h = \delta_j^h/h = (T_{hj}^{\frac{1}{2}} - T_{hj}^{-\frac{1}{2}})/h$$

$$\partial_{jj}^h = (\partial_j^h)^2 = (T_{hj} - 2 + T_{hj}^{-1})/h^2 \ .$$

We will use this notation with obvious modification: In two-dimensional problems, if (x,y) is used instead of (x_1,x_2), we will write ∂_x^h instead of ∂_1^h. In multi-grid descriptions where a sequence of grids is used with mesh sizes $h_1,\ldots,h_M$, we write ∂_j^k instead of $\partial_j^{h_k}$. When no confusion can arise, the superscript may also be omitted or replaced by a capital superscript denoting forward or central differencing:

$$\delta_j^F = T_{hj}^{\frac{1}{2}}\,\delta_j^h\ ,\quad \delta_j^B = T_{hj}^{-\frac{1}{2}}\,\delta_j^h\ ,\quad \delta_j^C = \mu_j^h\,\delta_j^h\ ,$$

$$\partial_j^F = T_{hj}^{\frac{1}{2}}\,\partial_j^h\ ,\quad \partial_j^B = T_{hj}^{-\frac{1}{2}}\,\partial_j^h\ ,\quad \partial_j^C = \mu_j^h\,\partial_j^h\ .$$

The finite-difference approximation to (3.1) can generally be written in the form

$$\sum_{\beta=1}^{q}\sum_{\underline{\nu}} b_{\alpha\beta\underline{\nu}}(\underline{x},h)U_\beta^h(\underline{x}+\underline{\nu}h) = F_\alpha^h(\underline{x})\ , \qquad (\underline{x}\in\Omega_\alpha^h\ ,\ \alpha=1,\ldots,q)\ . \tag{3.10}$$

Here $\sum_{\underline{\nu}}$ is a summation over a finite set of $\underline{\nu}=(\nu_1,\ldots,\nu_d)$, with ν_j usually being integers or half-integers. $b_{\alpha\beta\underline{\nu}}(\underline{x},h)$ are real functions, usually rational in h. Thus

$$\ell_{\alpha\beta}^h(\underline{x},h,\underline{T}_h) = \sum_{\underline{\nu}} b_{\alpha\beta\underline{\nu}}(\underline{x},h)\underline{T}_h^{\underline{\nu}} \tag{3.11}$$

will usually be the finite-difference approximation to $\ell_{\alpha\beta}(\underline{x},\underline{\partial})$. In some cases, however, the discrete system is not such a direct analog of the differential system. Sometimes, for example, even the number (q) of dependent functions in the differential system and in the discrete system are not the same. (The latter will be larger, e.g., when the difference equations are derived from high order finite-elements.) On the other hand, even when the discrete system is a direct analog of (3.1), obtained by replacing each derivative by a corresponding difference-quotient, and even if the differential system is elliptic, it does not follow that the discrete scheme is also "elliptic", i.e., the discrete system may be unstable. Thus, it is desired that the ellipticity of (3.10) be defined independently of the differential scheme it approximates.

We will sometimes write (3.10) in the compact form

$$L^h\,\underline{U}^h = \underline{F}^h\ , \tag{3.10'}$$

where $\underline{U}^h=(U_1^h,\ldots,U_q^h)^T$, $\underline{F}^h=(F_1^h,\ldots,F_q^h)^T$ and L^h is the $q\times q$ matrix $\ell_{\alpha\beta}^h$.

The *characteristic matrix* of system (3.10) is the $q \times q$ matrix

$$\ell^h(\underline{x},\underline{\theta}) = \{\ell^h_{\alpha\beta}(\underline{x},h,e^{i\underline{\theta}})\}_{\alpha,\beta=1,\ldots,q}$$

$$= \{\sum_{\underline{\nu}} b_{\alpha\beta\underline{\nu}}(\underline{x},h)e^{i\underline{\theta}\cdot\underline{\nu}}\}_{\alpha,\beta=1,\ldots,q} \quad .$$

and the *characteristic form* (or *symbol*) of (3.10) is defined by

$$\tilde{L}^h(\underline{x},h,\underline{\theta}) = \det \ell^h(\underline{x},\underline{\theta}) \quad , \quad (|\theta| \leq \pi) \quad , \tag{3.12}$$

where $\underline{\theta} = (\theta_1,\ldots,\theta_d)$, $e^{i\underline{\theta}} = (e^{i\theta_1},\ldots,e^{i\theta_d})$, $\underline{\theta}\cdot\underline{\nu} = \theta_1\nu_1 + \ldots + \theta_d\nu_d$, and $|\underline{\theta}| = \max[|\theta_1|,\ldots,|\theta_d|]$. The domain of Fourier components on the grid is restricted to $|\underline{\theta}| \leq \pi$, since for integer $\underline{\nu}$, $\exp(i\underline{\theta}\cdot\underline{\nu})$ and $\exp(i\underline{\theta}'\cdot\underline{\nu})$ coincide when $\underline{\theta} = \underline{\theta}'(\text{mod } 2\pi)$, i.e., when $(\theta_j-\theta_j')/2\pi$ are integers $(j = 1,\ldots,d)$. Note that $\underline{\theta}$ here corresponds to $h\underline{\Xi}$ of Section 3.1. Indeed $\ell^h_{\alpha\beta}$ is consistent with $\ell_{\alpha\beta}$ if and only if

$$\ell^h_{\alpha\beta}(\underline{x},h,e^{ih\underline{\Xi}}) \to \ell_{\alpha\beta}(\underline{x},i\underline{\Xi}) \quad \text{as} \quad h \to 0 \quad ,$$

and the order of approximating L by L^h is p if and only if

$$\ell^h_{\alpha\beta}(\underline{x},h,e^{ih\underline{\Xi}}) = \ell_{\alpha\beta}(\underline{x},i\underline{\Xi}) + O(h^p) \quad , \quad (1 \leq \alpha, \beta \leq q). \tag{3.12a}$$

We will assume that equations (3.10) are properly scaled, i.e., they are divided through by the proper power of h so that, for $\eta \to 0$ and fixed Ξ , $\tilde{L}^h(\underline{x},\eta,\eta\Xi)$ remains bounded while $\tilde{L}^h(\underline{x},\eta,\eta\Xi)/\eta$ is unbounded. The *order* $2m$ of the scheme (3.10) is defined as the highest order of h^{-1} in the terms of $\det(\ell^h_{\alpha\beta})$. That is, $2m$ is the lowest integer for which there exists a permutation $(i_1,\ldots,i_q)$ and component $|\underline{\theta}| \leq \pi$ such that

$$\lim_{\eta\to 0} \eta^{2m}\, \ell^h_{1i_1}(\underline{x},\eta,e^{i\underline{\theta}}) \quad \ldots \quad \ell^h_{qi_q}(\underline{x},\eta,e^{i\underline{\theta}}) \neq 0 \quad . \tag{3.13}$$

It is easy to see that if $\ell^h_{\alpha\beta}$ is consistent with $\ell_{\alpha\beta}$ then the order of (3.10) is the same as that of (3.1).

The *reduced symbol* of (3.10) is defined by

$$\overset{\vee}{L}{}^h(\underline{x},h,\underline{\theta}) = h^{2m}\,\tilde{L}^h(\underline{x},h,\underline{\theta}) \quad ,$$

and the *principal symbol* is defined as

$$\hat{L}^h(\underline{x},\underline{\theta}) = \overset{\vee}{L}{}^h(\underline{x},0,\underline{\theta}) \quad . \tag{3.14}$$

Note that $\hat{L}^h$ will be the symbol of the discrete approximation to the principal part of (3.1), i.e., the part represented in $\hat{L}$. Thus, another way of defining the principal symbol is as follows: There clearly exist integers $s_1,\dots,s_q$, $t_1,\dots,t_q$ such that, for every permutation $(i_1,\dots,i_q)$ that satisfies (3.13), the order of h^{-1} in $\ell^h_{\alpha i_\alpha}$ is $s_\alpha+t_{i_\alpha}$. Define the principal symbol of the term $\ell^h_{\alpha\beta}$ by

$$\hat{\ell}^h_{\alpha\beta}(\underline{x},\underline{\theta}) = \lim_{\eta\to 0} \eta^{s_\alpha+t_\beta} \sum_{\underline{\nu}} b_{\alpha\beta\underline{\nu}}(\underline{x},\eta)e^{i\underline{\theta}\cdot\underline{\nu}} \quad . \tag{3.14a}$$

(Note that for some terms the principal symbol may vanish.) Then the principal symbol (3.14) is also given by

$$\hat{L}^h = \det \hat{\ell}^h(\underline{x},\underline{\theta}) \quad , \tag{3.14b}$$

where $\hat{\ell}^h = \{\hat{\ell}^h_{\alpha\beta}\}$ is the $q \times q$ *principal characteristic matrix*.

3.3 T-Ellipticity

The concept of ellipticity for difference systems is more complicated than for differential ones. To distinguish between several possible variants of this concept we add prefixal letters to the adjective "elliptic". The system (3.10) is called *T-elliptic* at $\underline{x}$ if there exists a positive $\kappa(\underline{x})$ such that

$$|\hat{L}^h(\underline{x},\underline{\theta})| \geq \kappa(\underline{x})\ |\underline{\theta}|^{2m} \quad , \quad \text{for all} \quad |\theta| \leq \pi \quad . \tag{3.15}$$

This definition corresponds to (3.5a). If L^h is consistent with an *elliptic* differential operator L then the inequality holds automatically for small $|\theta|$, and hence (3.15) is equivalent to the weaker requirement

$$\hat{L}^h(\underline{x},\underline{\theta}) \neq 0 \quad , \quad \text{for all} \quad 0 < |\underline{\theta}| \leq \pi \quad , \tag{3.15a}$$

which is the analog of (3.5).

Thomée [T3] and Thomée and Westergren [TW] took (3.15a) as their definition of ellipticity for scalar equations ($q = 1$, $L^h = \ell_{11}^h$). This requirement by itself is not enough, however. For example, the scalar operator

$$L^h = \partial_{11}^h - T_{h2}\,\partial_{22}^h \tag{3.16}$$

has the symbol

$$\hat{L}^h(\underline{\theta}) = -(1 - \cos\theta_1) + e^{i\theta_2}(1 - \cos\theta_2)$$

which clearly satisfies (3.15a), but L^h is consistent with the hyperbolic differential operator $\partial_1^2 - \partial_2^2$ (the wave operator). In effect, however, definition (3.15) is used in [T3] and [TW]. An earlier version [T2] introduced the concept of ellipticity for symmetric scalar homogeneous operators with constant coefficients, i.e., for the case $q = 1$, $b_{11\underline{\nu}}(\underline{x},h) = h^{-2m}\hat{b}_{\underline{\nu}}$, $\hat{b}_{\underline{\nu}} = \hat{b}_{-\underline{\nu}}$. In this case if L^h is consistent with any differential operator L then T-ellipticity is equivalent to the requirement

$$\hat{L}^h(\underline{x},\underline{\theta}) > 0\,, \quad \text{for all} \quad 0 < |\theta| \leq \pi\,. \tag{3.17}$$

Operators satisfying (3.17) will be called here *symmetric-elliptic*.

Related definitions can be added here: The system (3.10) will be called *uniformly T-elliptic* in the domain Ω if its coefficients are uniformly bounded and there exists a positive constant κ independent of $\underline{x}$ for which (3.15) holds at all points $\underline{x} \in \Omega$. The largest such constant κ is called the *T-ellipticity constant*. The system (3.10) is called *semi-T-elliptic of order* $2m$ if

$$|\hat{L}^h(\underline{x},\underline{\theta})| \geq P_{2m}(\underline{x},\underline{\theta}) \quad \text{for all} \quad |\theta| \leq \pi \tag{3.18}$$

where P_{2m} is a non-negative homogeneous polynomial in $\underline{\theta}$ of order $2m$.

Using Fourier transformation it is easy to see that for homogeneous (principal part only) L^h with constant coefficients, the semi-T-ellipticity (3.18) is equivalent to the a priori estimate

$$\|P_{2m}(\underline{\partial}^h)\underline{u}^h\| \leq C\,\|L^h\underline{u}^h\|\,, \tag{3.19}$$

holding for all grid functions $\underline{u}^h$ vanishing outside a bounded domain. Here C is independent of $\underline{u}^h$ and h, and

$$\|\underline{u}^h\| = \sum_{j=1}^{q} \sum_{\underline{x}} |u_j(\underline{x})|^2 \quad , \tag{3.20}$$

the second summation being extended over all points $\underline{x}$ where $u_j(\underline{x})$ is defined. In particular, T-ellipticity is equivalent to an a priori estimate of all 2m-order derivatives of $\underline{u}^h$ in terms of the data $L^h\underline{u}^h$. Such estimates can then be extended further, using methods analogous to those used in the differential case (estimating lower-order derivatives of $\underline{u}^h$ in terms of higher-order derivatives of u^h and in terms of $\|\underline{u}^h\|$, perturbing from the constant-coefficients case to the continuous-coefficients case, etc.). For the scalar $(q = 1)$ case this is done in [TW]. Such methods, however, yield only interior estimates, similar to (3.7a). Estimates near boundaries, like (3.7), are much harder to get in the discrete case, where methods like coordinate transformation (used to transform boundaries to hyperplanes) are not applicable. Boundary estimates are available only for special classes of discrete elliptic problems (see [T5] and [T2]).

3.4 Quasi-ellipticity

In approximating elliptic systems, especially in the non-scalar $(q > 1)$ case, condition (3.15) is often violated only near $|\theta| = \pi$. This happens when central differencing is used to approximate odd-order derivatives in the principal part (e.g., central-differencing approximation to Cauchy-Riemann equations, see Sec. 5.2). Instead of (3.15), the weaker condition

$$|\hat{L}^h(\underline{x},\underline{\theta})| \geq \kappa(\underline{x})\left(\sum_{j=1}^{d} \sin^2\theta_j\right)^m \quad \text{for all real } \underline{\theta}, \tag{3.21}$$

is satisfied. Systems for which (3.21) holds will be called *quasi-T-elliptic*. If κ is independent of $\underline{x}$ the system is *uniformly quasi-T-elliptic*.

Using quasi-elliptic approximations, the results should be properly interpreted (sometimes they are not). Namely, the difference solution is meaningful only after proper averaging. Indeed, the operator is unstable for some Fourier components $\exp(i\underline{\theta}\cdot\underline{x}/h)$ for which $|\underline{\theta}| = \pi$ and $L^h(\underline{x},\underline{\theta}) = 0$. A small

perturbation (in F) in these (or in neighboring) high-frequency components may cause very large high-frequency changes in the solution (which corresponds to nothing of that sort in the differential case). The averaging should liquidate all such Fourier components. One such averaging is the operator

$$(\mu_1^h)^2 \ \dots \ (\mu_d^h)^2 = \sum_{|\underline{\nu}| \le 1} 2^{-d-|\nu_1|- \dots -|\nu_d|} T_h^{\underline{\nu}}$$

For properly averaged solutions of uniformly quasi-T-elliptic operators, stability results and a priori estimates can be obtained as for T-elliptic operators. But round-off errors may have here much larger effect, and the truncation errors will usually correspond to those obtained by T-elliptic operators on a coarser grid. Quasi-T-elliptic operators are therefore not recommended (but see Sec. 3.11).

3.5 S-ellipticity

The notion of T-ellipticity depends too much on vanishingly small meshsizes. One deficiency, for example, is that it allows operator translation. Namely, if L^h is T-elliptic, so does also $T_h^{\underline{\nu}} L^h$, for any fixed $\underline{\nu}$. This should certainly be avoided at finite mesh-sizes, since it would allow, for example, two difference equations at two neighboring points to contradict each other ($T_h^{\underline{\nu}} L^h U^h(\underline{x}) = F^h(\underline{x})$ contradicting $L^h U^h(\underline{x}+\underline{\nu}h) = F^h(\underline{x}+\underline{\nu}h)$ when $F^h(\underline{x}) \neq F^h(\underline{x}+\underline{\nu}h)$). This situation is avoided in [T3] and [TW] by requiring the coefficients $b_\nu(\underline{x},h)$ to be continuous in $\underline{x}$, so that the operator cannot have different translations at different points. This way of avoiding translations is not fully satisfactory since, in principle, we may like to use discontinuous coefficients in some problems; there is nothing basically wrong in certain discontinuities. Moreover, in this way translation is still allowed, although it is required to be the same translation at all points. This is inconvenient in the study of relaxation schemes, where a more definite relation is required between the location of an unknown and the location of the equation relaxed by it. To avoid translations, the following modification of the definition may be used.

As $\underline{\theta}$ changes continuously from $\underline{\theta}^\circ$ to $\underline{\theta}' \equiv \underline{\theta}^\circ (\mathrm{mod}\ 2\pi)$, the principal symbol $\hat{L}^h(\underline{x},\underline{\theta})$ traverses a closed curve (a loop) in the complex plane. The operator (3.10) is said to have <u>index</u> 0 if no such closed curve circles the origin (i.e., all such loops have index 0). The operator (3.10) is called *S-elliptic* (respectively *semi-S-elliptic, uniformly S-elliptic, quasi-S-elliptic*) if it is T-elliptic (respectively semi-T-elliptic, uniformly-T-elliptic, quasi-T-elliptic) and has index 0 . Note that (3.16) has index 0 , hence vanishing index is not enough for ellipticity. Note also that symmetric-T-elliptic operators are S-elliptic.

Every S-elliptic operator is of course T-elliptic. Conversely, *every T-elliptic operator can be translated to one and only one S-elliptic operator.* Indeed, if L_h is T-elliptic and if the index of the loop

$$\{L^h(\underline{x},\underline{\theta})\ ;\ 0 \leq \theta_j \leq \pi \text{ and } \theta_k \text{ is fixed for } k \neq j\}$$

is μ_j , it is easy to see that $T_h^{-\underline{\mu}} L^h$ is S-elliptic.

3.6 <u>R-ellipticity</u>

S-elliptic operators are inconvenient in constructing numerical approximations to differential equations. The following concept will be more useful.

The system (3.10) is called* *R-elliptic* iff

$$Re\ \hat{L}^h(\underline{x},\underline{\theta}) \geq \kappa(x)\ |\underline{\theta}|^{2m} , \quad \text{for all } |\underline{\theta}| \leq \pi , \tag{3.23}$$

uniformly R-elliptic iff $\kappa(x)$ is independent of x , *semi-R-elliptic* iff (cf. (3.18))

$$Re\ \hat{L}^h(\underline{x},\underline{\theta}) \geq P_{2m}(\underline{x},\underline{\theta}) , \quad \text{for all } |\underline{\theta}| \leq \pi , \tag{3.24}$$

and *quasi-R-elliptic* iff

$$Re\ \hat{L}^h(\underline{x},\underline{\theta}) \geq \kappa(x)\ \Big(\sum_{j=1}^{d} \sin^2\theta_j\Big)^m , \quad \text{for all } \underline{\theta} . \tag{3.25}$$

Note that if L^h is consistent with an elliptic operator L then L^h is R-elliptic iff

---

*The definition in [B5] is modified here. What we called there R-elliptic should more properly be called strongly-R-elliptic, since it is related to strongly-elliptic differential systems (see [ADN], page 43).

$$Re\ \hat{L}^h(\underline{x},\underline{\theta}) > 0\ ,\ \text{for all}\ \ 0 < |\underline{\theta}| \le \pi\ . \tag{3.26}$$

The condition (3.26), however, is not enough by itself. For example the operator

$$L^h = -\partial^h_{11} + h^2(\Delta^h)^2 = -\partial^h_{11} + h^2(\Sigma\ \partial^h_{jj})^2 \tag{3.27}$$

satisfies (3.26), but is only semi-R-elliptic.

R-ellipticity clearly entails S-ellipticity and hence allows no translation of the operator. On the other hand, all S-elliptic systems used in practice can easily be made R-elliptic by multiplying some of the equations by suitable constants. Every symmetric-elliptic operator consistent with any differential operator is clearly R-elliptic. But R-elliptic operators are not necessarily symmetric. An example is the asymmetric operator

$$L^h = -\partial_{11} - a\ T_{h,1}\ \partial_{22} \tag{3.28}$$

which is R-elliptic for $0 < a < 1$.

An important advantage of R-ellipticity is its additivity in the determinant. That is, in constructing the difference equations it is enough to construct each one of separate parts of $\det(\ell^h_{\alpha\beta})$ to be R-elliptic. In the scalar (q=1) case, in particular, the sum of R-elliptic operators is also R-elliptic, and hence R-elliptic operators can be constructed term by term. (See examples in [B5], Sec. 5.2, or Sec. 3.10 below.) In the non-scalar ($q > 1$) case, ellipticity is not additive: The sum of elliptic operators is not necessarily elliptic. But we can still exploit the additivity in the determinant. (See for example the constructions in Sections 5.2, 6.2, and 7.2).

There is a special case of ellipticity, called strong ellipticity ([ADN], p.43), which is additive even in the non-scalar case. The system (3.10) is called *strongly R-elliptic* (or strongly elliptic) if (i) $s_\alpha = t_\alpha$ (see Sec. 3.2), (ii) $(L^h u)_\alpha$ and u_α are defined on the same grid points, and (iii)

$$Re \sum_{\alpha,\beta=1}^{q} \hat{\ell}^h_{\alpha\beta}(\underline{x},\underline{\theta})\,\xi_\alpha \bar{\xi}_\beta \ge \kappa(x) \sum_{\alpha=1}^{q} |\underline{\theta}|^{2t_\alpha} |\xi_\alpha|^2 \tag{3.29}$$

for all complex $\underline{\xi} \neq 0$ and real $\underline{\theta} \neq 0$,

where $\{\hat{\ell}^h_{\alpha\beta}\}$ is the principal characteristic matrix (3.14a). Scalar R-elliptic operators are of course strongly elliptic. It is clear from (3.29) that the sum of strongly elliptic difference systems is also strongly elliptic. Hence, strongly-elliptic operators can be constructed term by term. On the other hand, important elliptic systems, such as Stokes and Navier-Stokes, and Cauchy-Riemann equations, are not strongly elliptic and thus cannot have strongly elliptic difference approximations.

Through Fourier transformation it is easy to see that, for homogeneous (principal part only) L^h with constant coefficients, strong ellipticity is equivalent to the a priori estimate

$$\sum_{\alpha=1}^{q} \sum_{j=1}^{d} \sum_{\underline{x}} \left| (\partial_j^h)^{t_\alpha} u_\alpha^h(\underline{x}) \right|^2 \leq C \sum_{\alpha} \sum_{\underline{x}} u_\alpha(\underline{x}) (L^h \underline{u})_\alpha(\underline{x}) \qquad (3.30)$$

holding for all grid function vanishing outside a bounded domain Ω . C is independent of h , u^h and Ω . For the scalar symmetric case, relation (3.30) is derived in [T2], leading to convergence theorems. Similar theorems can be derived for general strongly elliptic systems.

3.7 V-ellipticity

Suppose the solution U^h of (3.10), with suitable homogeneous boundary conditions, belongs to a normed linear space $\underline{V}^h$, with norm $\| \ \|$. Suppose also we can define the bilinear form

$$a^h(\underline{u}^h, \underline{v}^h) = \sum_{j=1}^{q} \sum_{\underline{x} \in \Omega_j^h} v_j^h(\underline{x}) (L^h \underline{u}^h)_j(\underline{x}) \ , \qquad (3.31)$$

$$(\underline{u}^h, \underline{v}^h \in \underline{V}^h) \ .$$

(This is of course possible only if v_j^h and $(L^h \underline{u}^h)_j$ are defined on the same set Ω_j^h of points $\underline{x}$.) The system (3.10) is called *V-elliptic* if a^h is continuous uniformly in h and there exists a positive constant α independent of h such that

$$a^h(\underline{u}^h, \underline{u}^h) \geq \alpha \|\underline{u}^h\|^2 \ , \text{ for all } \underline{u}^h \in \underline{V}^h \ , \qquad (3.32)$$

$$0 < h \leq h_0 \ .$$

Thus, in a suitable norm, strongly elliptic principal-homogeneous operators are V-elliptic (see (3.30)). The analysis of V-ellipticity is more developed than that of other ellipticity concepts, especially in finite element formulation (see [C1]). V-elliptic differential problems can be stated as variational problems. The usual procedure is to base the discretization on the variational form in a suitable finite-dimensional approximation space, so that the discrete problem is automatically V-elliptic and has some relations to the continuous problem which are very useful for theoretical analysis. On the other hand, V-ellipticity is not general enough, and important elliptic systems, like Cauchy-Riemann Stokes and Navier-Stokes, are not V-elliptic. The latter two can be reformulated as V-elliptic problems (see [T1]), but that reformulation is not suitable for the fast solution methods described below. (The solution process cannot stay in the divergence-free space $\underline{V}^h$, and explicit use of the pressure function is needed.) It is often the case that the most efficient finite-difference discretization of a V-elliptic problem is not quite V-elliptic itself. The construction of discrete approximations via the variational form is usually much more expensive in computer time and storage than direct differencing. Such an expense is tolerable when slow, storage-expensive solution processes, like elimination, are used. But when fast, storage-economical algebraic solvers, like the multi-grid solvers, are used, the usual finite-element assembly processes turn out to be by far the most expensive part of the calculations.

3.8 Scaled Ellipticity

For various purposes the above ellipticity definitions do not quite capture the stability properties we may be interested in. The definitions are given in terms of the principal part of the difference equations, and are therefore applicable to equations with lower-order terms only if the mesh-size h is "sufficiently small". In *differential* equations, the principal part indeed dominates the local behavior of solutions (whereas global behavior on large enough domains may be determined by the lower-order terms). In *difference* equations, by contr

unless the mesh-size h is "sufficiently small", the lower-order terms may dominate even the local behavior, since there exist no solution scales smaller than h. Only when h is small enough the principal part dominates the local behavior, and the above concepts and theory (e.g., [T3] and [TW]) applies. In many cases, however, those "sufficiently small" mesh-sizes are too small to be practical. In particular in singular perturbation operators, such as (3.9b), the principal part dominates only when the mesh-sizes h is small compared with the size (ε) of the perturbation (which in Navier-Stoke equations, for example, is proportional to the inverse of the Reynolds number). Furthermore, in such and other problems the form of L^h may depend on h: Central differencing may be used at sufficiently small h, while "upstream" differencing will be employed at larger h. The stability properties at $h \to 0$ are then clearly irrelevant for studying the schemes at larger values of h. Moreover, even at moderate values of ε, a theory for "sufficiently small" h will not be suitable for *multi-grid* schemes, where large values of h always participate in the solution process.

Thus, an improved definition of ellipticity for discrete equations must include lower-order terms. One way to handle this (see [B2], p.13) is to regard small parameters (like ε) in the difference equations as being functions of h (e.g., $\varepsilon = ah^s$). In particular, small coefficients $a_{\alpha\beta\underline{\gamma}}$ in the differential system (3.1) may be regarded as functions of h, or as scaled by h, namely

$$a_{\alpha\beta\underline{\gamma}} = \mathring{a}_{\alpha\beta\underline{\gamma}}\, h^{s_{\alpha\beta\underline{\gamma}}} \quad . \tag{3.33}$$

Differencing the differential equation in this form will change the dependence on h of the coefficients $b_{\alpha\beta\underline{\nu}}(\underline{x},h)$ in (3.10), so we can write them as new functions $\mathring{b}_{\alpha\beta\underline{\nu}}(\underline{x},h)$. This will then change the dependence on h of $\ell^h_{\alpha\beta}$ and $\tilde{L}^h$ and the order $2m$ of the system, which will hence be denoted by $\mathring{\ell}^h_{\alpha\beta}$, $\mathring{L}^h$ and $2\mathring{m}$, respectively. The *scaled principal symbol* will be defined by

$$\overset{\circ}{L}{}^{h}(\underline{x},\underline{\theta}) = \lim_{\eta\to 0} \eta^{2\overset{\circ}{m}}\, \tilde{\overset{\circ}{L}}{}^{h}(\underline{x},\eta,\underline{\theta}) \quad . \tag{3.34}$$

The system (3.10) will be called *scaled T-elliptic* at $\underline{x}$ if there exists positive $\kappa(x)$ such that

$$\left|\overset{\circ}{L}{}^{h}(\underline{x},\underline{\theta})\right| \geq \kappa(\underline{x})\ |\underline{\theta}|^{2\overset{\circ}{m}} \quad , \qquad \text{for all} \quad |\theta| \leq \pi \quad . \tag{3.35}$$

Employing $\overset{\circ}{L}{}^{h}$ instead of $\hat{L}^{h}$, we can similarly define *scaled semi-T-elliptic, scaled quasi-T-elliptic, scaled S-elliptic, scaled semi-S-elliptic, scaled quasi-S-elliptic, scaled R-elliptic, scaled semi-R-elliptic, scaled quasi-R-elliptic, scaled strongly elliptic* and *scaled V-elliptic* operators. Uniform ellipticity of all these kinds is similarly defined, with "ellipticity constants" κ independent of $\underline{x}$.

As a simple illustration, consider the central-differencing approximation to (3.9b)

$$L^{h} = -\varepsilon \Sigma\, \partial^{h}_{jj} + \Sigma\, a_j \mu^{h}_{j} \partial^{h}_{j} \tag{3.37}$$

whose symbol is

$$\tilde{L}^{h}(\underline{\theta}) = \frac{4\varepsilon}{h^2} \Sigma \sin^2 \frac{\theta_j}{2} + \frac{i}{h} \Sigma\, a_j \sin\theta_j \quad . \tag{3.38}$$

The principal symbol is

$$\hat{L}^{h}(\underline{\theta}) = 4\varepsilon \Sigma \sin^2 \frac{\theta_j}{2} \quad . \tag{3.39}$$

For the scaling $\varepsilon = \eta h$, the scaled principal operator is

$$\overset{\circ}{L}{}^{h}(\underline{\theta}) = 4\eta \Sigma \sin^2 \frac{\theta_j}{2} + i \Sigma\, a_j \sin\theta_j \quad , \tag{3.40}$$

so that $\mathrm{Re}\ \overset{\circ}{L}{}^{h}(\underline{\theta}) > (4\eta/d\pi^2)|\underline{\theta}|^2$, and the operator is scaled R-elliptic. For any scaling $\varepsilon = o(h)$, however, the symbol is

$$\overset{\circ}{L}{}^{h}(\underline{\theta}) = i \Sigma\, a_j \sin\theta_j \quad ,$$

which can clearly vanish for various $0 < |\underline{\theta}| \leq \pi$. Hence (3.37) is not scaled elliptic for $\varepsilon = o(h)$.

Difference approximations to (3.9b) which are R-elliptic uniformly in ε are described below (Section 3.10).

Scaled ellipticity is very useful in discussing approximations to singular-perturbation operators and their multigrid solutions [B5]. A thorough analysis of scaled T-elliptic

difference approximations to scalar (q=1) singular perturbation problems has been carried out by Frank [F1]-[F6]. He writes the difference equations in terms of the singular-perturbation parameter ε (properly defined) and uses the scaling $\varepsilon = \rho^{-1}h$. Furthermore, he shows [F4] that ellipticity and coerciveness conditions are both necessary and sufficient for certain a priori estimates to hold uniformly in $0 < \rho < \infty$. Also defined is a "weak" ellipticity concept which guarantees uniform stability only in an interval $0 < \rho_1 \leq \rho \leq \rho_2 < \infty$. The a priori estimates are only proved for the infinite domain ($\Omega = \mathbb{R}^d$) or for a bounded one-dimensional (d=1) domain, (i.e., an interval), but the ellipticity concept has of course wider applicability. It applies, however, only to the case where both the perturbed differential operator L and the reduced operator (i.e., the lower-order operator obtained from L by dropping the higher-order perturbation) are elliptic. Unfortunately most singular perturbations in fluid dynamics, and even simple problems like (3.9b) (for $d > 1$), are not of this kind.

3.9 Measures of discrete ellipticity: Stability of high-frequencies

Note in example (3.37) with the scaling $\varepsilon = \eta h$, that for any fixed η the operator is formally scaled-elliptic, no matter how small η is. Indeed, there is no critical-value of η below which the operator starts to be "bad". (For $\eta < \eta_o = \frac{1}{2}\max(|a|,|b|)$ the operator is no longer of "positive type", but the discretization errors for $\eta = .9\eta_o$ are essentially the same as for $\eta = 1.1\,\eta_o$, contrary to a common belief. See [F4].) It is clear, however, that for $\eta \ll \max(|a|,|b|)$ the scheme (3.38) behaves as badly as a non-elliptic scheme. Namely, small high-frequency perturbations (in $L^h u^h$) cause unduly large (even though bounded) high-frequency changes in u^h. Thus, for practical purposes, what we should really be interested in is not only *whether* L^h is elliptic or not, but mainly *how much* "elliptic" it is *at a specific value of* h; i.e., by *how much* high-frequency modes (modes with wavelengths comparable to h) are unduly magnified by $(L^h)^{-1}$. Lower frequency modes are usually

taken care of by the consistency of L^h with an elliptic operator L. Indeed, some such modes may necessarily be unstable, since they approximate unstable modes of L. (Unstable modes exist in indefinite operators L, such as (3.9d) with large enough w.)

Thus, in some analog to (3.8), we can define the *hT-ellipticity measure* of L^h at $\underline{x}$ as

$$E_T^h(L^h,\underline{x}) = C \min_{\rho\pi\le|\underline{\theta}|,|\underline{\theta}'|\le\pi} \left| \frac{\tilde{L}^h(\underline{x},h,\underline{\theta})}{\tilde{L}^h(\underline{x},h,\underline{\theta}')} \right| , \tag{3.43a}$$

where $|\underline{\theta}| = \max|\theta_j|$. We choose the normalization $C = 2d/(1-\cos\rho\pi)$, so that for the five-point Laplace operator we get $E_T^h = 1$. The choice of ρ is somewhat arbitrary. For multi-grid purposes a natural ρ is the mesh-size ratio h_k/h_{k-1}, because the Fourier components $\exp(i\underline{\theta}\cdot\underline{x}/h_k)$ in the high-frequency range $(h_k/h_{k-1})\pi \le |\underline{\theta}| \le \pi$ are exactly those modes on grid h_k which are not "visible" (i.e., they alias with lower modes) on the coarser grid h_{k-1}. Hence we will take hereinafter

$$\rho = \frac{1}{2} , \quad C = 2d . \tag{3.43b}$$

The hT-ellipticity measure of L^h in a domain Ω^h is defined as

$$E_T^h(L^h) = E_T^h(L^h,\Omega) = \min_{\underline{x}\in\Omega} E_T^h(L^h,\underline{x}) . \tag{3.44}$$

Various other, equivalent measures for the local (scale h) ellipticity could be similarly introduced. For R-elliptic operators L^h, a useful measure will be the *hR-ellipticity measure*

$$E_R^h(L^h,\underline{x}) = 2d \min_{\frac{\pi}{2}\le|\underline{\theta}|,|\underline{\theta}'|\le\pi} \frac{Re\ \tilde{L}^h(\underline{x},h,\underline{\theta})}{Re\ \tilde{L}^h(\underline{x},h,\underline{\theta}')} . \tag{3.45}$$

The precise value of E^h is of course not important, but its order of magnitude is a very significant property of L^h. We will say that the difference system L^h has good h-ellipticity if $E^h(L^h,\underline{x})$ are not small (compared with 1). It implies stability of the high-frequency modes.

The h-ellipticity measures are defined for a specific mesh-size, the very mesh-size chosen for the actual computations. It should be emphasized that $E^h(L^h) > 0$ does not imply ellipticity of L^h. In fact, L^h may be consistent with a non-elliptic differential operator and still have a good h-ellipticity measure. An example is the hyperbolic oeprator (3.16). Conversely, some elliptic operators will necessarily have bad h-ellipticity measures for some (large enough) values of h. An example is any difference approximation to (3.9d) for mesh-sizes $h = O(\omega^{-1})$. Indeed, at such values of h, usual difference equations cannot produce good approximations to (3.9d), since the grid does not resolve the natural oscillations of the continuous solution (whose wavelength is $2\pi/\omega$).

The one case in which small h-ellipticity measures for all values of h should give no trouble is when L^h approximates a degenerate elliptic differential operator L, such as (3.9a), which itself has a small ellipticity measure $E(L,\underline{x}) = \varepsilon$. The usual $O(h^2)$ approximation to (3.9a),

$$L^h = -\varepsilon\partial^h_{11} - \partial^h_{22} - \ldots - \partial^h_{dd} \quad , \tag{3.46}$$

indeed has $E^h(L^h,\underline{x}) = O(\varepsilon)$. It is of course possible to construct $O(h^2)$ approximations to (3.9a) which have good h-ellipticity. For example,

$$L^h = -\varepsilon\partial^h_{11} + h^2(\partial^h_{11})^2 - \partial^h_{22} - \ldots - \partial^h_{dd} \quad . \tag{3.47}$$

But (3.46) is not necessarily worse than (3.47), since its high-frequency instability reflects a similar behavior of the differential operator. (A nice multi-grid possibility is to use (3.47) in relaxation (if pointwise relaxation is desired) and (3.46) in the residual transfers. See Sec. 3.11.)

In fact, all we need from L^h is that its h-ellipticity measure be good in the same sense in which the measure of L is good. For degenerate operators this means good *semi-h-ellipticity measures*. The semi-hR-ellipticity measures, for example, are defined by

$$E^h_R(L^h,\underline{x};\Phi) = C \min_{\substack{\frac{\pi}{2}\le|\underline{\theta}|,|\underline{\theta}'|\le\pi \\ \underline{\theta},\underline{\theta}'\in\Phi}} \frac{Re\ \tilde{L}^h(\underline{x},h,\underline{\theta})}{Re\ \tilde{L}^h(\underline{x},h,\underline{\theta}')} \quad , \tag{3.48}$$

where Φ is a suitable subspace. In case of approximating (3.9a), for example, Φ should be the subspace $\{\theta_1 = 0\}$, since it is the largest Φ yielding

$$\mathcal{A}(L,x;\Phi) = \lim_{Q\to\infty} \min_{\substack{\|\Xi\| = \|\Xi'\| = Q \\ \Xi,\Xi' \in \Phi}} \left| \frac{L(\underline{x},\Xi)}{L(\underline{x},\Xi')} \right| = 1 .$$

3.10 Construction of elliptic difference systems

The two main considerations in selecting difference approximations to a given differential operator are accuracy and stability. Accuracy reflects the quality of the approximation for smooth components, i.e., for Fourier components $\exp(i\,\Xi\cdot\underline{x})$ with $\Xi \ll h^{-1}$. Its most significant measure is the order of approximation p (see (3.12a)). On the other end of the spectrum, Fourier components with $|\Xi| > \pi/h$ are not visible on the grid and are not approximated at all. Even the highest frequencies that *are* visible cannot have a good approximation, and there is no point in trying too hard to approximate them, since slightly higher frequencies are not approximated anyway. All we need in the high frequency range ($\pi/(2h) \lesssim \Xi \lesssim \pi/h$, say) is that the difference solutions cannot be much larger than the differential ones. For this all we need is the high-frequency stability of the difference operator, discussed above (Sec. 3.9).

To construct p-order approximations to (3.1), one can simply approximate each $\ell_{\alpha\beta}(\underline{x},\underline{\partial})$ by $\ell^h_{\alpha\beta} = \ell_{\alpha\beta}(\underline{x},\underline{\partial}^{(p)})$ where $\partial_j^{(p)}$ is any p-order approximation to ∂_j, not necessarily the same at all occurrences. Most such approximations will not be stable. So the problem at hand is *how to construct such p-order approximations which will have also good h-ellipticity measures* (although unstable approximations can be used in multi-grid processes; cf. Sec. 3.11).

3.10.1. Central approximations of order $p = 2$ are widely used. For general even p, the simplest (i.e., containing minimal number of points) p-order central approximations to the first derivative ∂_j and the second derivative $-\partial_{jj}$ are given by

$$\partial_j^{[p]} = \partial_j^h \sum_{k=0}^{p/2-1} \frac{1}{2k+1} A_k \left(-\frac{1}{4}\delta_{jj}^h\right)^k \tag{3.49}$$

or

$$\partial_j^{(p)C} = \partial_j^C \sum_{k=0}^{p/2-1} B_k \left(-\frac{1}{4}\delta_{jj}^h\right)^k \tag{3.50}$$

and

$$-\partial_{jj}^{[p]} = -\partial_{jj}^h \sum_{k=0}^{p/2-1} \frac{1}{k+1} B_k \left(-\frac{1}{4}\delta_{jj}^h\right)^k \quad , \tag{3.51}$$

where $A_0 = B_0 = 1$, $A_k = (1 - \frac{1}{2k})A_{k-1}$, $B_k = (1 + \frac{1}{2k})^{-1} B_{k-1}$, and ∂_j^h , ∂_j^C , ∂_{jj}^h and δ_{jj}^h are defined in Sec. 3.2. The first missing term in each summation, evaluated at some intermediate point, gives the local truncation error (see [B5]). Note that (3.49) is centered at half-way between grid lines, while (3.50) and (3.51) are centered on grid lines. Odd-order central approximations do not exist; the simplest formulae always yield the next (higher) even order.

The symbols corresponding to $h\partial_j^h$, $h\partial_j^C$, $-h^2\partial_{jj}^h$ and $-\frac{1}{4}\delta_{jj}^h$ are, respectively, $2i \sin(\theta_j/2)$, $i \sin\theta_j$, $4 \sin^2(\theta_j/2)$ and $\sin^2(\theta_j/2)$. Hence it is clear that the symbol corresponding to $\partial_j^{[p]}$ is $i\Xi_j'$, where $h\Xi_j' = \theta_j + O(\theta_j^{p+1})$ is real and does not vanish in the relevant range $0 < |\theta_j| \leq \pi$. Suppose we can use the difference scheme

$$\ell_{\alpha\beta}^h = \ell_{\alpha\beta}(\underline{x}, \underline{\partial}^{[p]}) \quad , \qquad (\alpha,\beta = 1,\ldots,q) \quad . \tag{3.52}$$

The discrete symbol $\tilde{L}^h(\underline{x},h,\underline{\theta})$ will then coincide with the differential symbol $\tilde{L}(\underline{x},\Xi_j')$, where Ξ_j' vanishes if and only if θ_j does. Hence (3.52) preserves all the ellipticity properties of the differential system. Moreover, every degeneracy in L^h will reflect a similar degeneracy in L . For k-order derivatives $(\partial_j)^k$ the approximation (3.52) uses $k(p-1)+1$ grid points, whereas the most compact p-order approximation needs $k+p-1$ points. Hence, for $k > 1$, (3.52) is most compact only for $p = 2$. It can usually be replaced by the most compact operator, without destroying the ellipticity properties. Indeed, the most compact approximation to $(\partial_j)^k$ is obtained from $(\partial_j^{[p]})^k$ by dropping all differences of order higher than $k+p-2$. It is clear from (3.49) that

all the terms dropped have in the symbol the same sign as the remaining terms.

The main trouble with (3.52) is that it cannot always be used. For homogeneous (principal term only) operators without mixed terms, the grids can often be "staggered" so that (3.52) is applicable (see examples in Secs. 5.2 and 6.2). Consider, however, the approximation of the scalar operator $-\varepsilon\partial_{jj}+a\partial_j$. The approximation (3.52) to the first term is centered at grid lines, while for the second term the centering is half-way between grid lines, which is a contradiction. The central approximation of this operator, and of (3.9b) and similar operators, must employ (3.50) together with (3.51). The trouble here is that for large values of ah/ε this operator does not have good h-ellipticity measures: $\tilde{L}^h(x,h,\pi) \to 0$ as $ah/\varepsilon \to \infty$. Thus, as long as the terms using (3.51) outweigh those of (3.50), the operator is stable. But the stability practically disappears (for $\theta_j = \pi$) for large ah/ε.

3.10.2. Upstream differencing. Consider a general scalar ($q = 1$) differential operator L. A convenient way of constructing p-order R-elliptic approximations L^h to L, with good hR-ellipticity measures, is to construct separately a p-order Semi-R-elliptic approximation, with good semi-hR-ellipticity measures, to each term in L (except of course for those mixed-derivative terms which cannot have R-elliptic approximations. If such derivatives are present, however, there must also be present non-mixed derivatives of the same order to make up for it). Assume the difference equations are centered at grid points. Then the simplest p-order central approximations, such as (3.50) and (3.51), are semi-R-elliptic, and would give R-elliptic L^h whenever L is elliptic. The h-ellipticity measure may, however, be bad (like in (3.37), for large $h\,\Sigma|a_j|/\varepsilon$). Indeed, the simplest central p-order approximations to any even (non-mixed) derivative, such as (3.51), all have good semi-hR-ellipticity measure. But the corresponding approximations to the odd derivatives, such as (3.50), have measure 0. (Good semi-hT-ellipticity measure would not help, since it is not additive.)

Good hR-ellipticity measures for approximations of odd derivatives can be obtained by adding to the central approximation a dissipative term of high enough order and suitable magnitude. For example, p-order approximation to the first derivative is obtained by adding to (3.50) any O(1) positive multiple of either $h^{-1}(-\delta^h_{jj})^{p/2}$ or $\partial^B_j(-\delta^h_{jj})^{p/2}$ or $-\partial^F_j(-\delta^h_{jj})^{p/2}$. All these terms have good hR-ellipticity measures. The first of them uses the same grid points used by (3.50), but reduce the order of approximation by 1. The latter two retain the approximation order p. The choice of the positive multiple, and the choice between ∂^B_j and $-\partial^F_j$, can be based on the desire to have most compact formula, i.e., to have p-order approximations to ∂_j based on p+1 grid-points. With such choices the p-order approximations then turn out to be

$$\partial_j^{(p)U} = \partial_j^C \sum_{k=0}^{[p/2]} B_k(-\tfrac{1}{4}\delta^h_{jj})^k + \frac{2}{h} B_{[p/2]}(-\tfrac{1}{4}\delta^h_{jj})^{[(p+1/2]} \sigma_j \quad , \tag{3.53}$$

where $\sigma_j = s_j$ for odd p, $\sigma_j = h\partial^C_j - \frac{1}{2}\delta^h_{jj} s_j$ for even p and s_j is the sign of the coefficient of the approximated ∂_j. (This sign should of course be taken from the R-elliptic form of the differential equation, e.g., the form in which ∂_{jj} terms have negative coefficients). It is easy to see that $s_j\partial_j^{(p)U}$ is R-elliptic and has good semi-hR-ellipticity measure.

For $p=1$ and $p=2$ the operators (3.53) are completely one-sided, e.g., using backward differencing when the coefficient of ∂_j is positive. Since in applications that coefficient usually represents the j-component of some velocity, such differencing is called *upstream* (or *"upwind"*) differencing. (Hence the superscript U in (3.53).) Note, however, that for $p > 2$ the operators are not one-sided; some downstream grid-points are also used. The one-sided p-order operators based on p+1 grid-points are not R-elliptic (for $p > 2$): their use would produce unstable computations.

3.10.3. Artificial viscosity. Instead of upstream differencing, one can use central differencing and just add a general dissipative term, of suitable order and magnitude, not attached to any particular differential term. The simplest such term is either $C(h)\Sigma_j(-\delta^h_{jj})^{p'}$ or $C(h)(-\Sigma_j\delta^h_{jj})^{p'}$, where $C(h) > 0$ is comparable in magnitude to the coefficients of the central difference operator, and p' is just high enough to maintain the desired approximation order. This procedure is often simpler than upstream differencing since here the difference formula does not depend on the signs of various coefficients. Also, the h-ellipticity measure is in this way guaranteed to be good, while in upstream differencing it may have degeneracies. For example, in upstream approximation to (3.9b), degeneracy occurs when some a_j are much smaller than others. In fact, the artificial viscosity is exactly equivalent to upstream differencing in case the grid directions happen to be such that all a_j are equal.

The artificial viscosity (whether explicit or through upstream differencing) makes it possible to treat singular perturbation problems even when the reduced problem is not elliptic (e.g., (3.9b) for large $h\Sigma|a_j|/\varepsilon$, or Navier Stokes equations for large $Rh|\underline{U}|$), in contrast to the cases treated in [F4]-[F6]. Such problems usually have thin transition layers, like boundary layers, turning point, shocks, etc. The introduction of artificial viscosity causes these layers to be smeared over several mesh-sizes (and hence resolvable by the grid). Multi-level *adaptive* techniques would use finer levels around such layers, thus making them as thin as needed (see [B5]). In this way the total artificial viscosity added to the system can be made small, since away from such layers the solution U is smooth and the artificial viscosity terms are therefore small (their magnitude being $O(h^p)$ relative to other terms).

3.10.4. Divergence forms. The differential equations often include terms like $-\partial_j(a\partial_j)$. This form is called divergence form or conservative form, since it usually results from physical conservation laws. It is best to discretize the term directly in this form (rather than in the form $-a\partial_{jj}-(\partial_j a)\partial_j$), because that will produce difference schemes with conservation

properties similar to the differential schemes and hence with guaranteed convergence (see [LW]). Also (for elliptic systems) the divergence-form discretization can conveniently be made in terms of central differencing only. Indeed, using (3.49) for any even p we have the compact p-order central approximation to $-\partial_j(a\partial_j)$

$$(-\partial_j(a\partial_j))^{[p]} = [-\partial_j^{[p]}(a\partial_j^{[p]})]_p \quad , \tag{3.54}$$

where $[\ ]_p$ denotes the removal of all terms of order higher than $O(h^p)$ resulting from the product. (3.51) is the special case $a \equiv 1$. For $p=2$ (3.54) is the familiar operator $-\partial_j^h(a\partial_j^h)$. For $p=4$,

$$(-\partial_j(a\partial_j))^{[4]} = \partial_j^h(a - \tfrac{1}{24}(\delta_{jj}^h a + a\delta_{jj}^h))\partial_j^h \quad . \tag{3.55}$$

In the sense of [LW], (3.54) is always conservative, since ∂_j^h is a common left-factor in all its terms. Also, (3.54) has a good semi-hR-ellipticity measure (in the sense of (3.48), where $\Phi = \{\theta_i = 0 \text{ for } i \neq j\}$, and for smooth a).

At points away from the boundary, the less compact formula $-\partial_j^{[p]}(a\partial_j^{[p]})$ may actually be simpler and more efficient to evaluate than (3.54). This formula is of course also conservative and with good semi-hR-ellipticity measure.

3.10.5. High order approximations near boundaries. The various formulae above, in particular the higher-order (i.e., $p>2$ if central, $p>1$ otherwise) or the non-compact ones, are often inapplicable near boundaries, where not enough neighboring grid points are available. We need then to replace some points in the difference operator by others, maintaining the same approximation order. This can generally be done by adding $O(h^p)$ terms to the operator. Terms which are semi-R-elliptic are preferable, where possible, but they do not seem to be necessary. Generally, the theoretical requirements near boundaries are far less clear, and require further investigation.

Another possibility is to use lower-order approximations near boundaries, using a finer grid there to make up for the lower accuracy. The multi-level adaptive technique [B5] will

do it automatically when the restriction on the approximation order is imposed. For certain error norms, lower-order operators near boundaries can be used without grid refinement and without spoiling the global order of approximation (cf. [BH]). The best possibility may be the combined use of low-order elliptic and high-order non-elliptic approximations, as in Sec. 3.11.

3.10.6. Non-scalar systems. The construction of p-order R-elliptic approximations with good hR-ellipticity measure to non-scalar ($q > 1$) operators can again be done term by term, except that the terms now are those in $\tilde{L}$ (the determinant of the matrix operator). See example in Sec. 5.2. When $\tilde{L}$ is a product of elliptic operators, one can construct separately approximations to terms of each of these operators. See examples in Sec. 6.2 and 7.2.

3.11 Multi-level differencing

An important feature of the multi-level fast solvers is the effective separation between the treatment of high-frequency modes (modes with $\frac{1}{2}\pi \leq |\underline{\theta}| \leq \pi$, affected only by relaxation) and low-frequency ones (affected mainly by the coarse-grid corrections; the lower the mode, the less its amplitude is changed by a relaxation sweep). This feature can be exploited in various ways. For example, the conflict between using more accurate central differencing or the corresponding, less accurate but more stable, upstream differencing has a simple multi-level resolution. Upstream differencing is much better for the highest frequencies and should therefore be used *in relaxation*. The central differencing is better for lower modes (the lower the mode, the better it is) hence it should be used in the residuals transfer (i.e., L^{k+1} of (2.13), when $k+1 = \ell$).

This procedure will ensure stability (and hence also efficient smoothing; see Sec. 4.2) together with the higher-order accuracy of central differencing. Note that such a multi-level process will not converge to zero residuals, since it uses two conflicting difference schemes. The very point is, indeed, that the solution produced is a better approximation to the *differential* solution than can be produced by

either scheme.

Generally, the global approximation order p of the multi-level scheme will be determined by the order of the difference operator used in the residuals transfer. This operator need not be stable. It is only in relaxation that a stable operator (i.e., with good hR-ellipticity measure) should be employed, and this operator can be of lower approximation order.

Observe that the lower order operator can be used on the *coarser* grids both for relaxation and for residuals transfers, since those grids act only as correction grids. Thus L^{k+1} in (2.10) should be of the higher order only for $k+1 = \ell$.

The τ-extrapolation technique (Sec. 2.4) can, in fact, be regarded as a special case of this procedure. There, the higher-order operator of the residuals transfer is in effect constructed as a combination of lower-order operators on two levels, which is simpler to program.

4. LOCAL MODE ANALYSIS OF MULTI-LEVEL PROCESSES

An important feature of multi-grid solvers is that their computational work can fully be predicted by local mode (Fourier) analysis. This is an analysis applied to general nonlinear problems in the following way: The difference equations are linearized around some approximate solution, and the coefficients of the linearized equations are frozen at local values. (Or, more generally, the coefficients may assume some typical mode of oscillation. See Sec. 4.7.1.) The resulting constant-coefficient (or single-mode-coefficients) problem is then assumed to hold in a grid covering the entire space, and its convergence properties under various processes can be studied in terms of the Fourier components of the error. This local analysis is a very good approximation to the true behavior of modes with short wavelengths, which interact at short distances and are therefore not influenced by distant boundaries and slow changes of coefficients. It is inaccurate for long modes, but those may be ignored in the multi-grid work estimates, since long-modes convergence is obtained on coarser grids, where the computational work is negligible.

Indeed, the predictions of this analysis turns out to be very precise, so much so that they can be used in developing the programs (see Sec. 6.5 for example). As long as the cycling algorithm (see Sec. 2.6) does not attain the convergence factor $\hat{\mu}$ (see Secs. 4.4 and 4.6), it must contain a programming bug or a conceptual error. Such errors are very common with inexperienced multi-gridders, especially in their treatment of boundary conditions, hence it is recommended that codes be gradually developed, starting from simplest cases and insisting at each stage on attaining the theoretical convergence factor.

The local mode analysis should of course be supplemented by some other considerations. These are discussed in Sec.4.3.

4.1 Smoothing factors

The simplest (and most useful) application of the local mode analysis is to compute the error-smoothing power of a given relaxation scheme. We assume that the relaxation is *consistently ordered*. This means that its sweeps consist of passes in each of which the order of two points $\underline{x}$ and $\underline{y}$ (whether $\underline{x}$ is relaxed before, after or simultaneously with $\underline{y}$) depends only on $\underline{x}-\underline{y}$.

There are many types of relaxation, especially for non-scalar systems. The simplest one can be called *strongly pointwise Gauss-Seidel relaxation*. In it, to each finite difference equation there corresponds one and only one discrete unknown. The relaxation scan the equations by some order. Each equation in its turn is satisfied by changing the value of the corresponding unknown. This relaxation is always convergent if and only if the discrete system of equations is positive definite (upon assigning a proper sign to each equation). Hence this type of relaxation is natural for V-elliptic systems, or for strongly elliptic systems (with sufficiently small mesh-sizes, in case non-principal terms are present).

For elliptic systems which are not strongly elliptic there is no natural one-to-one correspondence between equations and unknowns. (This is already true in the differential system. See Sec. 5.3, for example.) More natural

then is what we call *pointwise Collective Gauss-Seidel (CGS)* relaxation. For this type of relaxation it is assumed that the grid is not staggered. The q unknown functions and the q differential equations are all defined on the same grid-points so that at each point q unknowns and q difference equations are centered. The relaxation sweep consists of scanning these grid points in some order. At each point in its turn the system of q difference equations is satisfied by simultaneously changing the q unknowns. A slightly more general method of relaxation, called pointwise Collective Successive Over-Relaxation (CSOR), is to change at each point the q unknowns by changes equal to the CGS changes multiplied by some "relaxation parameter" ω. (More generally, ω may be a $q \times q$ matrix.) Denoting by $\underline{u}$ and $\bar{\underline{u}}$ the approximate solution before and after such a relaxation sweep, respectively, we get the relation

$$\sum_{\underline{\nu}\in N^-} B_{\underline{\nu}}\, \underline{u}(\underline{x}+\underline{\nu}h) + \sum_{\underline{\nu}\in N^+} B_{\underline{\nu}}\, \bar{\underline{u}}(\underline{x}+\underline{\nu}h) + \\ + B_{\underline{0}}\left[\left(1-\frac{1}{\omega}\right)\underline{u}(\underline{x}) + \frac{1}{\omega}\,\bar{\underline{u}}(\underline{x})\right] \\ = \underline{F}^h(\underline{x}) \quad , \tag{4.1}$$

where $B_{\underline{\nu}}$ is the $q \times q$ matrix $b_{\alpha\beta\underline{\nu}}$ (see (3.10)), N^+ is the set of neighborhood indices $\underline{\nu}$ such that $(\underline{x}+\underline{\nu}h)$ is scanned before the point $\underline{x}$ and hence new value $\bar{\underline{u}}$ are already used at those points, and N^- is the complementary set of indices such that $(\underline{x}+\underline{\nu}h)$ is scanned after $\underline{x}$. Denoting by $\underline{v} = \underline{U}^h - \underline{u}$ and $\bar{\underline{v}} = \underline{U}^h - \bar{\underline{u}}$ the error before and after the sweep, respectively, we get from (4.1) and (3.10)

$$\sum_{\underline{\nu}\in N^-} B_{\underline{\nu}}\, \underline{v}(\underline{x}+\underline{\nu}h) + \sum_{\underline{\nu}\in N^+} B_{\underline{\nu}}\, \bar{\underline{v}}(\underline{x}+\underline{\nu}h) + \\ + B_{\underline{0}}\left[\left(1-\frac{1}{\omega}\right)\underline{v}(\underline{x}) + \frac{1}{\omega}\,\bar{\underline{v}}(\underline{x})\right] = 0 \; . \tag{4.2}$$

In the local mode analysis we assume $\underline{v}$ to be defined in the entire space, and hence it can be expanded in the discrete Fourier expansion

$$\underline{v}(\underline{x}) = \int \hat{\underline{v}}(\underline{\theta})\, \exp(i\, \underline{\theta}\cdot\underline{x}/h)\, d\underline{x} \; . \tag{4.3}$$

Similarly

$$\bar{\underline{v}}(\underline{x}) = \int \bar{\hat{\underline{v}}}(\underline{\theta})\, \exp(i\, \underline{\theta}\cdot\underline{x}/h)\, d\underline{x} \; . \tag{4.4}$$

It is also assumed in this analysis that B_ν are independent of $\underline{x}$ and hence, upon substituting (4.3) and (4.4) in (4.2),

$$B_- + B_0(1 - \frac{1}{\omega}) + [B_+ + \frac{1}{2} B_0]R = 0 \quad , \tag{4.5}$$

where $R = R(\underline{\theta})$ is the *relaxation amplification matrix*, i.e., $\hat{\underline{v}}(\underline{\theta}) = R(\underline{\theta})\ \hat{\underline{v}}(\theta)$, and

$$B_- = \sum_{\underline{\nu} \in N^-} B_{\underline{\nu}}\, e^{i\underline{\theta}\cdot\underline{\nu}} \quad , \quad B_+ = \sum_{\underline{\nu} \in N^+} B_{\underline{\nu}}\, e^{i\underline{\theta}\cdot\underline{\nu}} \quad .$$

The *relaxation amplification factor* $\mu(\underline{\theta})$ is the eigenvalue of $R(\underline{\theta})$ with the largest magnitude. The *smoothing factor* $\bar{\mu}$ is defined as the worst (largest) magnitude of amplification factors of high-frequency components:

$$\bar{\mu} = \max_{\frac{\pi}{2} \le |\underline{\theta}| \le \pi} |\mu(\theta)| \quad . \tag{4.6}$$

The *smoothing rate* is $\log(1/\bar{\mu})$.

In other types of relaxation, calculating the smoothing factors may be more complicated, since in addition to u and $\bar{u}$ there may be several intermediate values of the approximate solution (see for example Secs. 5.3 and 6.3 below). There will then be several equations like (4.5), instead of just one, from which R should be eliminated, or the amplification factor $\mu(\underline{\theta})$ should be computed for any desired $\underline{\theta}$. A general computer routine, called SMORATE, has been developed for this purpose. The user inputs the details of the relaxation scheme, and the routine outputs the smoothing factor $\bar{\mu}$, a map of the amplification factors $|\mu(\theta)|$, as well as various other related information, including certain estimates of the convergence factors (see Secs. 4.4 and 4.5).

4.2 Construction of Relaxation Schemes

With the SMORATE routine we can *evaluate* and *optimize* relaxation schemes over some range of possibilities. On the other hand, it does not provide us with a general method of *constructing* good schemes. For non-scalar problems the construction is not trivial at all. Although CSOR usually provides smoothing factors bounded away from 1 (i.e., $\bar{\mu} < 1$ and $\bar{\mu}$ does not tend to 1 as $h \to 0$), more "natural" schemes may provide much better factors (see Secs. 5.3, 6.3,

7.3. Note there that for such schemes it may be easier to calculate $\bar{\mu}$ in terms of the residual function $L^h V$ rather than in terms of the error function V. Both of course yield the same amplification matrix R).

The construction of good relaxation schemes, like the construction of good finite difference equations, depends on some physical intuition and expertise that can be derived from considering the simplest cases. See Sec. 3.2 in [B2], Sec. 3.3 in [B3], Sec. 6 in [B5], and Secs. 5.3, 6.3 and 7.3 below. Here we emphasize some general considerations, relating smoothing to the h-ellipticity measures.

A necessary condition that pointwise relaxation schemes for the difference system L^h *can be devised with good smoothing factors is that* L^h *has a good hT-ellipticity measure* $E_T^h(L^h)$. Indeed, if for some mode $\underline{\theta}$ the value of $|\tilde{L}^h(h,\underline{\theta})|$ is small (compared to its values for other modes), then the error $V(x) = A \exp(i\,\underline{\theta}\cdot\underline{x}/h)$ has small residuals (compared with the residuals of other errors with the same amplitude A), and can therefore have only small (compared with A) corrections, no matter what relaxation scheme is used (as long as it is a pointwise scheme, where the correction is determined only by the local residuals). This is in fact the reason why relaxation is not efficient for small $\underline{\theta}$ (smooth error components), where $\tilde{L}^h(h,\underline{\theta}) = O(1)$, compared with $\tilde{L}^h = O(h^{-2m})$ for some high-frequencies. Small hT-ellipticity measure means that $|\tilde{L}^h(h,\underline{\theta})|$ is small for some high-frequency $\underline{\theta}$, hence the error mode $\underline{\theta}$ is little affected by any pointwise relaxation, hence $|\mu(\underline{\theta})|$ is close to 1, hence $\bar{\mu}$ is close to 1.

The above necessary condition may also be sufficient (in the context of constant-coefficient equations discussed here). As an indication, consider the CSOR scheme mentioned above. Writing $\mu(\theta) = 1 - \tilde{\mu}(\theta)\omega$, it follows from (4.5) that

$$\det\{B - \tilde{\mu}\, B_0 - \tilde{\mu}\,\omega\, B_+\} = 0\ , \tag{4.7}$$

where $B = B_+ + B_- + B_0$ is the characteristic matrix $\ell^h(\underline{\theta})$ (cf. Sec. 3.2). Hence $\mu = 1-\lambda\omega + O(\omega^2)$, where λ is a root of the equation

$$\det\{B - \lambda B_0\} = 0 \quad . \tag{4.8}$$

Thus, for sufficiently small ω , $|\mu|$ is smaller than 1 (and bounded away from 1 as $h \to 0$) iff $Re\ \lambda > 0$ (and $Re\ \lambda$ is bounded from 0) for all roots λ of (4.8). In the scalar case this means that, for sufficiently small ω and all sufficiently small h , $|\mu(\underline{\theta})|$ is bounded away from 1, for all $|\underline{\theta}|$ bounded away from 0 , if and only if L^h is R-elliptic (cf. [B2]). For any *fixed* h , $\bar{\mu} < 1$ for sufficiently small ω if and only if $E_R^h(L^h) > 0$; and, moreover, *large values of* $E_R^h(L^h)$ *imply large values of* $\bar{\lambda}$, *where* $\bar{\mu} = 1 - \bar{\lambda}\omega + O(\omega^2)$.

In case the h-ellipticity measure $E_T^h(L^h)$ is small, so that no pointwise relaxation would have good smoothing rates, one can still use *block relaxation*, such as line relaxation, or (if $d \geq 3$) plane relaxation, etc. Generally, if Φ is some subspace, then a Φ-relaxation is any relaxation where, simultaneously with $\underline{x}$, we relax all the points $\underline{y}$ such that $\underline{y} - \underline{x} \in \Phi$. For example, Φ-CGS is a relaxation where all the q equations at all the points $\underline{y}$ such that $\underline{y}-\underline{x} \in \Phi$ are satisfied simultaneously by changing all the unknowns corresponding to these points. Φ-CSOR is similar, with changes which are the Φ-CGS changes multiplied by the relaxation parameter ω . For this type of relaxation it is not necessary to have good $E_T^h(L^h)$, but it is necessary to have good semi-h-ellipticity measure $E_T^h(L^h,\Phi)$ (see (3.48)).

Block relaxation will thus be used for degenerate elliptic difference operators like (3.46). For (3.47) we can use pointwise relaxation, despite the degeneracy of the differential operator. Generally, the degeneracy that require block relaxation is such that the system of difference equations can be decomposed into blocks of equations with only weak inter-block couplings. These are the blocks that should be taken simultaneously in the block relaxation. For example, in (3.46) the blocks are the unknowns corresponding to the hyperplanes $\{x_1 = \text{const.}\}$, which become decoupled as $\varepsilon \to 0$. Hence pointwise relaxation would not smooth the error: High-frequency functions of x_1 will not be affected by relaxation. They will, however, be efficiently reduced by the hyperplane

relaxation. Another example of such degeneracy occurs in (upstream) approximations to (3.9b), when some of the a_j are much smaller than others. A similar situation arises in Navier Stokes equations with large Reynolds number when the velocity direction approximately coincides with a grid-line or a grid plane.

Another alternative in treating such degeneracies is to still use pointwise relaxation but to employ as the multi-grid coarser level a grid which is *coarser only in the directions of smoothing*, i.e., only inside the blocks (only in the $x_2,\ldots,x_d$ directions, in the above example).

A similar approach (a multi-grid remedy for certain inefficiencies in relaxation) can also be taken in solving quasi-elliptic equations (Sec. 3.4). Relaxation there is inefficient for the unstable error components ($|\underline{\theta}| \approx \pi$ with small $L^h(\underline{\theta})$). These components are suitably averaged out by (3.22). So the multi-grid remedy for the inefficient smoothing is to use (3.22) in transferring the residuals to the coarse grid. As a result, the multi-grid process will have fast convergence for the *averaged* solution, which is the only meaningful solution.

4.3 Supplementary considerations

Some discrepancies between real computations and the local mode analysis should be taken into account. Most important is to realize that the error smoothing process in relaxation does not continue indefinitely. Except for some ideal cases, a certain level of high-frequency errors is always coupled to the smooth errors. Starting from a completely smooth error function, a certain level of high-frequency error modes is generated by the relaxation sweeps because of interaction with boundaries and variations in the coefficients of the finite-difference equations. This level of *coupled nonsmoothness* will persist as relaxation slows down. Further relaxation sweeps will be wasteful. Moreover, if the error is smoother than this level, relaxation may even magnify the high-frequency errors instead of reducing them, and it is best to avoid relaxing altogether (cf. analysis in Sec. A.2 of [B3]).

The practical rule, at any rate, is always the same. Continue relaxation as long as it exhibits fast convergence rate. When it slows down, switch to coarse grids -- the error is always sufficiently smooth for that purpose. In case of highly-oscillatory coefficients, the residual function is not smooth, however, and should be transferred to the coarse grid by full weighting.

Another consideration to supplement the local mode analysis is the relaxation of boundary conditions. These should not be scrambled together with the interior equations -- their smoothing is a separate process, and there is no way of transferring to coarser level the residual of an equation which is a combination of interior and boundary difference equations. Also, care should be taken that the boundary relaxation does not disturb too much the interior smoothness. For example, if a second-order partial differential equation with Neumann boundary conditions is given, with some smooth initial error, and if we change the solution near boundaries so as to satisfy the Neumann conditions, then we introduce interior residuals near the boundary which are much larger than (in fact, $O(h^{-1})$ times) the other interior residuals. Generally, imposing non-Dirichlet boundary conditions will similarly seriously impair the interior smoothness. The effect of this will not be serious if full residual weighting (i.e., residual transfer to the coarse grids such as (3.22) which assigns the proper total weight to the residual at each fine-grid point) is used near the boundary. Better still, this trouble can be completely avoided if we note that we need not impose the boundary condition in relaxation. All we need is to smooth its error, with a smoothing factor as good as the interior smoothing factor. In the above example, this is obtained if, at each boundary point in its turn, instead of satisfying the boundary condition there, we change its error to become the average of the errors at adjacent boundary points. In case of a smooth error function, this procedure would introduce only $O(h)$ (instead of the above $O(h^{-1})$) relative disturbance to the interior residual function.

4.4 Multi-grid convergence factors: One level analysis

The main purpose of the local mode analysis is to predict the convergence rate of multi-grid cycling algorithms (Sec. 2.6). The simplest prediction, the one-level analysis, is in terms of the smoothing factor $\bar{\mu}$. We assume that the relaxation sweeps over level k affect error components $\exp(i\,\underline{\xi}\cdot\underline{x})$ only in the range $\pi/h_{k-1} \le |\underline{\xi}| \le \pi/h_k$ (see Sec. 6.2 in [B3]). One such sweep reduces such a component by $\mu(\underline{\xi}h_k)$. Hence, if the multi-grid cycle includes s sweeps over each level, all the error components will be reduced at worst by $\bar{\mu}^s$. We assume of course that s is not too large (see Sec. 4.3) and that the interpolation and residual transfers are of the proper orders (see Appendix A in [B3]).

Let us denote by $\overset{\circ}{\mu}$ the convergence factor per relaxation work unit, where the work unit is the work of one sweep over the finest level M (so that a sweep on level k costs $2^{d(k-M)}$ work units), and where all other work (such as the work of interpolations and transfers) is ignored. The above cycle with s sweeps on each level costs roughly $s(1+2^{-d} + 2^{-2d} + \ldots) \approx s(1-2^{-d})^{-1}$ such work units. Hence the one-level analysis estimate is

$$\overset{\circ}{\mu} \approx \bar{\mu}^{(1-2^{-d})} \quad . \tag{4.9}$$

This estimate is somewhat crude, but it is easy to obtain (e.g., using SMORATE – see Sec. 4.2) and it turns out in practice to be at most 20% off the accurate two-level estimate.

4.5 Mean square convergence factors

Estimate (4.9) is sometimes too pessimistic. It predicts the convergence of the worst component, as implied by (4.6). Sometimes, except for a very small range of $\underline{\theta}$, the values $|\mu(\underline{\theta})|$ are all considerably smaller than $\bar{\mu}$, and hence the worst convergence rate (4.9) will usually become dominant only after many cycles. The actual decrease of the error in a small number of cycles depends then on the relative magnitude of the various components in the initial error. Denoting by $\hat{V}(\underline{\theta})$ the amplitude of the $\underline{\theta}$ component in the initial error, its amplitude after several cycles is $\mu(\underline{\theta})^s\,\hat{V}(\underline{\theta})$, $(\pi/2 \le |\underline{\theta}| \le \pi)$, where s is the total number of sweeps on

the finest level made during these cycles. Hence the L_2 norm of the high-frequency errors is reduced by the factor

$$\mu_s = \left\{ \frac{\int |\hat{V}(\underline{\theta}) \cdot \mu(\underline{\theta})^s|^2 d\underline{\theta}}{\int |\hat{V}(\underline{\theta})|^2 \, d\underline{\theta}} \right\}^{\frac{1}{2}} , \tag{4.10}$$

where the integrations are over $\frac{\pi}{2} \leq |\underline{\theta}| \leq \pi$. Hence, the convergence factor per relaxation-work-unit is given by

$$\overset{\circ}{\mu}_s \approx (\mu_s)^{s^{-1}(1-2^{-d})} , \tag{4.11}$$

where we assume that the additional work on coarser grids (to obtain convergence similar to (4.10) also in the low frequencies) is still about $(2^d-1)^{-1}$ times the work on the finest grid. It is easy to see that asymptotically, as $s \to \infty$, estimate (4.11) indeed tends to (4.9).

An example of a relaxation scheme for which most $|\mu(\theta)|$ are much smaller than $\bar{\mu}$ is given by the pointwise Gauss-Seidel relaxation of (3.46), where $\bar{\mu} = |\mu(\pi/2,0,\ldots,0)| \approx 1-2\varepsilon/(d-1)$. Numerical experiments are reported in [P1] in which the initial error was random, so that $\hat{V}(\underline{\theta}) \equiv 1$ could be assumed in (4.10). The prediction (4.11), for various values of s , turned out to be very precise.

The computer routine SMORATE (see Sec. 2) calculates both estimate (4.11) and the asymptotic estimate (4.9). Its user can input the initial distribution $\hat{V}(\underline{\theta})$. Otherwise $\hat{V}(\underline{\theta}) \equiv 1$ is assumed, corresponding to random initial errors.

4.6 Multi-grid convergence factors: Two-level analysis

The one-level analysis is not always accurate enough for debugging purposes (comparisons with numerical experiments). More importantly, it yields no information concerning the inter-level operations (interpolations, residual transfers), and hence no tool for optimizing them. The two-level analysis presented below takes into account all the operations between the finest level M and the next coarser level $M-1$, and makes some approximating assumptions concerning the still coarser levels (M-2, M-3, etc.).

The Fourier mode $\exp(i\theta \cdot x/h_M)$ of level-M errors appears on level $M-1$ as the mode $\exp(i2\underline{\theta} \cdot \underline{x}/h_{M-1})$, since

$h_{M-1} = 2h_M$. Hence, on level M-1 it coincides with every mode $\exp(i\,\theta'\cdot x/h_M)$ such that $\underline{\theta}' \equiv \underline{\theta}$ (mode π) . Thus, the inter-level analysis introduces coupling between each *lower mode* $\underline{\theta}$ $(0 \leq |\underline{\theta}| \leq \frac{\pi}{2})$ and its (2^d-1) high-frequency *harmonics* $\{\underline{\theta}' : \frac{\pi}{2} \leq |\underline{\theta}'| \leq \pi\ ,\ \underline{\theta}' \equiv \underline{\theta}(\text{mod } \pi)\}$. In particular, the corrections interpolation I_{M-1}^M is represented by the $q \times q$ matrices $\hat{I}_{M-1}^M(\underline{\theta}')$ defined by

$$I_{M-1}^M \hat{V}\, e^{i\underline{\theta}\cdot\underline{x}/h} = \sum_{\theta'\equiv\theta\,(\text{mod}\pi)} \hat{I}_{M-1}^M(\underline{\theta}')\, \hat{V}\, e^{i\underline{\theta}'\cdot\underline{x}/h} \quad , \tag{4.12}$$

where $\hat{V}$ is any q-vector and $0 < |\theta| \leq \pi/2$. $\hat{I}_{M-1}^M(\underline{\theta}')$ is called the *symbol* of I_{M-1}^M . Usually $\hat{I}_{M-1}^M$ is diagonal, i.e., any two correction functions v_α^{M-1} and v_β^{M-1} are interpolated independently of each other. In higher-order finite element formulations, however, one discrete function may represent a derivative of another, and the interpolation of each will then involve the other. Most often used is the universal I-order multi-polynomial interpolation, for which

$$\hat{I}_{M-1}^M(\underline{\theta}')_{\alpha,\beta} = \delta_{\alpha\beta} \prod_{j=1}^{d} \varphi_I(\cos\theta_j') \quad , \tag{4.13}$$

where $\varphi_2(\xi) = (1+\xi)/2$, $\varphi_4(\xi) = (8 + 3\xi + 5\xi^3)/16$, etc. For staggered grids $\varphi_I(\xi)$ depends on the relative positions of the coarse and the fine grids, which may depend on α .

The residual transfers I_M^{M-1} are similarly represented by a symbol $\hat{I}_M^{M-1}(\underline{\theta}')$, where

$$I_M^{M-1} \hat{V}\, e^{i\underline{\theta}'\cdot\underline{x}/h} = \hat{I}_M^{M-1}(\underline{\theta}')\, \hat{V}\, e^{i\underline{\theta}\cdot\underline{x}/h} \quad . \tag{4.14}$$

For non-staggered grids the right-hand side of (4.13) can describe these matrices, too, including also the case $\varphi_0(x) \equiv 1$, which is called "injection".

We consider a multi-grid cycle which includes s relaxation sweeps on level M , transferring then the residuals to level M-1 , solving this residual problem on level M-1 and interpolating its solution as a correction to level M . (A more realistic cycle is discussed in Sec. 4.7.2 .) In terms of the relaxation amplification matrix $R(\underline{\theta})$ (see Sec. 4.2), the characteristic matrices $\ell^M = \ell^h$ and $\ell^{M-1} = \ell^{2h}$ (see Sec. 3.2), and the interpolation symbols introduced above, the

$2^d q \times 2^d q$ *amplification matrix* of the cycle can be written as

$$C(\theta) = [\overset{v}{I} - \overset{vM}{I}_{M-1}(\theta)\ \ell^{2h}(2\theta)^{-1}\ \overset{vM-1}{I}_M(\theta)\ \overset{vh}{\ell}(\theta)]\ \overset{v}{R}(\theta)^s \ . \quad (4.15)$$

This matrix transforms the $(2^d q)$-vector of error amplitudes

$$\overset{v}{V}(\underline{\theta}) = (\hat{\underline{V}}(\underline{\theta}^1),\ldots,\hat{\underline{V}}(\underline{\theta}^{2^d}))^T \ ,$$

where $\underline{\theta}^1,\ldots,\underline{\theta}^{2^d}$ are the harmonics of $\underline{\theta}$, and $\hat{\underline{V}}(\underline{\theta}^j)$ is the q-vector amplitude of the $\underline{\theta}^j$ error mode. The matrices on the right-hand side of (4.15) are block matrices defined by

		dimension
$\overset{v}{R}(\underline{\theta})$	$= \mathrm{diag}\{R(\underline{\theta}^1),\ldots,R(\underline{\theta}^{2^d})\}$	$2^d q \times 2^d q$
$\overset{vh}{\ell}(\underline{\theta})$	$= \mathrm{diag}\{\ell^h(\underline{\theta}^1),\ldots,\ell^h(\underline{\theta}^{2^d})\}$	$2^d q \times 2^d q$
$\overset{vM-1}{I}_M(\underline{\theta})$	$= (\hat{I}_M(\underline{\theta}^1),\ldots,\hat{I}^{M-1}_M(\underline{\theta}^{2^d}))$	$q \times 2^d q$
$\overset{vM}{I}_{M-1}(\underline{\theta})$	$= (\hat{I}^M_{M-1}(\underline{\theta}^1),\ldots,\hat{I}^M_{M-1}(\underline{\theta}^{2^d}))^T$	$2^d q \times q$

and $\overset{v}{I}$ is the $2^d q \times 2^d q$ identity matrix. The two-level amplification factor $\lambda(\theta)$ is the largest (in magnitude) eigenvalue of $C(\theta)$. The *two-level convergence factor* is

$$\bar{\lambda} = \sup_{0<|\theta|\le\pi/2} |\lambda(\theta)| \ . \quad (4.16)$$

For various cases $\bar{\lambda}$ was calculated by suitable computer routines. A general routine for this purpose (similar to SMORATE) is not yet available.

To estimate $\overset{\circ}{\mu}$, the convergence factor per relaxation-work-unit, we assume again that the same number of relaxation sweeps is made on all levels (cf. Sec. 4.7.2) and hence

$$\overset{\circ}{\mu} \approx \bar{\lambda}^{s^{-1}(1-2^{-d})} \ . \quad (4.17a)$$

Here, only the work of relaxation is counted. When residuals are transferred to the coarse grid not by injection, this transfer roughly costs another work-unit, and a more accurate estimate then is

$$\overset{\circ}{\mu} \approx \bar{\lambda}^{(s+1)^{-1}(1-2^{-d})} \ . \quad (4.17b)$$

Numerical values for $\bar{\lambda}$ and $\overset{\circ}{\mu}$ are included in Sec. 4.8.

4.7 Multi-grid factors: Additional remarks

4.7.1. Oscillatory coefficients. To study the influence of non-constant coefficients, we can consider the extreme case of adding high-frequency oscillations. Instead of constant $b_{\alpha\beta\underline{\nu}}$ (cf. (3.10)), we can take

$$b_{\alpha\beta\underline{\nu}}(\underline{x},h) = \sum_{j=1}^{2^d} b_{\alpha\beta\underline{\nu}j} \exp(i\,\underline{\theta}^j\cdot\underline{x}/h) \quad , \tag{4.18}$$

where $\underline{\theta}^j$ are the harmonics of $\underline{\theta}$ (cf. Sec. 4.6). The theory for this case will go through as in Sec. 4.6, except that $\overset{\vee}{R}(\underline{\theta})$ and $\overset{\vee h}{\ell}(\underline{\theta})$ will no longer be block-diagonal, but full matrices. The coarse grid symbol ℓ^{2h} remains the same, since the coarse grid operator is assumed to have constant coefficients. (Indeed, when the fine-grid coefficients (4.18) are transferred to the coarse grid, either by injection or by averaging, one gets a constant-coefficient coarse-grid operator. Incidentally, numerical experiments [O1] have shown that, averaging is much preferable to injection in case the high-frequency harmonics in (4.18) have large amplitudes.)

4.7.2. Perturbations. The cycle described in Sec. 4.6 is not fully realistic. In the real cycle the residual problem on level M-1 is not completely solved, it is only solved to a relative tolerance δ (see Step F in Sec. 2.2). We can get a good idea about the difference between the ideal cycle and the real one by replacing $C(\theta)$ with

$$C_\eta(\underline{\theta}) = [\overset{\vee}{I} - \overset{\vee M}{I}_{M-1}(\underline{\theta})(1+\eta)\ell^{2h}(2\underline{\theta})^{-1}\overset{\vee M-1}{I}_M(\underline{\theta})\overset{\vee h}{\ell}(\underline{\theta})]\overset{\vee}{R}(\underline{\theta})^s \tag{4.19}$$

and replacing $\bar{\lambda}$ by

$$\bar{\lambda}_\delta = \max_{|\eta|\le\delta} \sup_{0<|\theta|<\pi/2} \left|\lambda_\eta(\underline{\theta})\right| \quad , \tag{4.20}$$

where $\lambda_\eta(\underline{\theta})$ is the largest eigenvalue of $C_\eta(\underline{\theta})$. Various values of δ could be used, but for consistency with the above work assumption (i.e., that the cycle contains the same number of sweeps on all levels) one should take $\delta = \bar{\lambda}_\delta$. This equation may require iterative solution, but two iterations are actually enough, namely, we can use the estimate $\bar{\lambda} = \bar{\lambda}_{\bar{\lambda}_0}$. We can also *optimize* δ *and* s by minimizing $\mathring{\mu} \approx \delta^{1/w}$, where w is defined by $\bar{\lambda}_\delta^w = \delta^{s+w/4}$ (or $\delta^{s+1+w/4}$).

4.7.3. Rigorous upper bounds for multi-grid convergence rates in the L_2 norm (for constant-coefficients problems in the infinite space) can be derived by slight changes in the above estimates. See an example in Appendix C of [B3]. Such rigorous bounds are not too unrealistic: the rigorous bound for $\log \mathring{\mu}$ is typically 3 times the real asymptotic value.

4.7.4. Realistic asymptotic convergence factors are given by (4.16) and (4.20). The difference between them is only a few percent. The asymptotic convergence exhibited by cycling algorithms [D2] deviates from (4.16) less than (4.20) does. The shape of the domain (and its being finite) proves experimentally ([S1],[O1]) to have no effect on the asymptotic (i.e., the worst) convergence factor, except when level M is very coarse. This validates a basic assumption of the local mode analysis.

In case of a nonlinear or variable-coefficient problem, the real convergence factor should be no worse than the various factors obtained by local mode analyses at all points of the domain, and the analysis of Sec. 4.7.1. There is not enough experimental results to generally confirm this, but in various cases ([S1], [D2]) it proved true.

4.7.5. Precise comparisons. For debugging purposes (see Sec. 2.6), a precise agreement is desired between the theoretical and experimental asymptotic convergence rates. The comparison then should not be made between values of $\mathring{\mu}$, such as (4.17), since those values already involve some imprecise assumption about the work on coarser grids. (The theoretical $\mathring{\mu}$ is used for theoretical optimizations.) Instead, the theoretical values (4.16) and (4.20) should be compared with the experimental values of $\bar{\lambda}$. The comparison can be made even more precise by comparing (4.16) with an experimental $\bar{\lambda}$ obtained by a slightly modified algorithm, in which much smaller δ is employed, at least for level $M-1$. To obtain the asymptotic (i.e., the worst possible) experimental rate $\bar{\lambda}$ without spending too many cycles, it is helpful to start with initial errors devised to contain a large amplitude of a worst component (a component $\underline{\theta}$ for which the sup in (4.16) is

attained or approached). Another alternative is to compute and compare two-level mean-square convergence factors, similar to the one-level mean-square factors of Sec. 4.5.

4.7.6. Simplified multi-grid analysis, which separately treat the relaxation process and the coarse-grid-correction process, is described in Appendix A of [B3]. It is less precise than the above two-level analysis, but it is good enough for algorithmic optimizations, and gives a clear idea of the interpolation orders that should be used, and other parameters. In fact, general rules for the inter-level operations emerge from order-of-magnitude considerations, and all that is left to be decided in every particular problem is the relaxation scheme. This decision can be based on the smoothing-rate analysis (Secs. 4.1, 4.2, 4.3) alone.

4.8 Numerical tables

Smoothing factors and the corresponding one-level convergence-rate predictions are given as Table 1 in [B3]. One-level mean-square convergence factors for degenerate operators are given in [P1] where they are compared with numerical experiments. Two-level convergence factors for Stokes equations are given in Sec. 6.5 below. The following table is part of more extensive tables to appear in [D2].

TABLE 4.1

Two-level convergence prediction for the 5-point Poisson equation with Gauss-Seidel relaxation.

δ	s sweeps per cycle	Residuals Injection (I=0)		Residual Weightings (I=2)	
		$\bar{\lambda}_\delta$	$\mathring{\mu}=\bar{\lambda}_\delta^{.75/s}$	$\bar{\lambda}_\delta$	$\mathring{\mu}=\bar{\lambda}_\delta^{.75/(s+1)}$
0	1	.4472	.5469	.4000	.7092
	2	.2000	.5469	.1923	.6622
	3	.0894	.5469	.1183	.6701
	4	.0416	.5509	.0833	.6888
	5	.0278	.5843	.0640	.7092
.1	2	.2314	.5776	.1821	.6532
	3	.0996	.5618	.1047	.6550
	4	.0981	.6471	.0955	.7031
.2	2	.2619	.6050	.1952	.6647
	3	.1971	.6663	.1924	.7342
	4	.1962	.7369	.1934	.7816
.3	3	.2957	.7374	.2926	.7942

5. CAUCHY-RIEMANN EQUATIONS

5.1 The differential problem

As a first simple example of an elliptic system we have studied the equations

$$U_x + V_y = F_1 \tag{5.1a}$$

$$U_y - V_x = F_2 \tag{5.1b}$$

in a domain Ω, where $U = U(x,y)$ and $V = V(x,y)$ are the unknown functions, the subscripts denote partial derivatives, and $F_i = F_i(x,y)$ are given functions. All functions are real. The homogeneous system $F_1 \equiv F_2 \equiv 0$ are the usual Cauchy-Riemann equations, which express analyticity of the complex function $U+iV$.

The matrix-operator form of (5.1) is

$$L\begin{pmatrix} U \\ V \end{pmatrix} \equiv \begin{pmatrix} \partial_x & \partial_y \\ \partial_y & -\partial_x \end{pmatrix} \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \tag{5.2}$$

where ∂_x and ∂_y are partial derivatives with respect to x and y, respectively. The determinant of L is the Laplace operator $-\Delta = -\partial_x^2 - \partial_y^2$. Hence (5.2) or (5.1) is a second-order elliptic system, and its solution is determined by one condition along the boundary $\partial\Omega$. As such a boundary condition we can, for example, require

$$(U(x),V(x))_n = G(x) \quad , \quad (x \in \partial\Omega) \ , \tag{5.3}$$

where $(U,V)_n$ denotes the component of the vector (U,V) normal to the boundary in the outward direction. From (5.1a), (5.3) and the divergence theorem we get the "compatibility condition"

$$\int_\Omega F_2 dxdy = \int_{\partial\Omega} G \, ds \ . \tag{5.4}$$

If (5.4) holds then equations (5.1) or (5.2), with the boundary condition (5.3), is a well-posed problem: A unique solution exists and depends continuously on the data F_1, F_2, and G.

5.2 Discrete Cauchy-Riemann equations

Suppose we first try to approximate (5.1) by the central difference equations

$$\frac{U^h(x+h,y)-U^h(x-h,y)}{2h} + \frac{V^h(x,y+h)-V^h(x,y-h)}{2h} = F_1^h(x,y) \qquad (5.5a)$$

$$\frac{U^h(x,y+h)-U(x,y-h)}{2h} - \frac{V^h(x+h,y)-V^h(x-h,y)}{2h} = F_2^h(x,y). \qquad (5.5b)$$

The corresponding difference operator is

$$L^h = \begin{pmatrix} \mu_x^h \partial_x^h & \mu_y^h \partial_y^h \\ \mu_y^h \partial_y^h & -\mu_x^h \partial_x^h \end{pmatrix} \qquad (5.6)$$

so that

$$\det L^h = -(\mu_x^h \delta_x^h)^2 - (\mu_y^h \delta_y^h)^2 \quad , \qquad (5.7)$$

with the symbol (see Sec. 3.2)

$$\hat{L}^h(\theta_1,\theta_2) = \sin^2\theta_1 + \sin^2\theta_2 \ . \qquad (5.8)$$

This operator is not elliptic, since $\hat{L}^h(\pi,0) = \hat{L}^h(0,\pi) = \hat{L}^h(\pi,\pi) = 0$.

Indeed, the homogeneous (5.5) equations ($F_1^h \equiv F_2^h \equiv 0$) have the oscillatory solutions

$$\begin{aligned} U^h(\alpha h,\beta h) &= C_1(-1)^\alpha + C_2(-1)^\beta + C_3(-1)^{\alpha+\beta} \\ V^h(\alpha h,\beta h) &= C_4(-1)^\alpha + C_5(-1)^\beta + C_6(-1)^{\alpha+\beta} \end{aligned} \qquad (5.9)$$

which do not approximate any solution of the corresponding differential equation. Note, however, that solutions like (5.9) vanish *in the average*, i.e., $M^h U^h = M^h V^h = 0$ for a suitable local averaging operator M^h. For example, suitable averaging operators are $M^h = \mu_x^h \mu_y^h$ or $M^h = (\mu_x^h \mu_y^h)^2$. (In the first case the grid-lines of $M^h u^h$ are half-way between grid lines of u^h). Generally, the solutions of (5.5) will be good solutions in the average. Such difference operators we call *quasi elliptic*. See Sec. 3.4.

Let us now construct an elliptic difference approximation L^h to (5.1). If the equations are to have the form

$$D_x^1 U^h + D_y^2 V^h = F_1^h \tag{5.10a}$$

$$D_y^3 U^h - D_x^4 V^h = F_2^h \tag{5.10b}$$

where D_x^j and D_y^j are some difference approximations to ∂_x and ∂_y, then $\det L^h = -D_x^1 D_x^4 - D_y^2 D_y^3$ should be an elliptic approximation to the Laplace operator $-\Delta$. The simplest such operator is the five-point operator which is obtained by taking either

$$D_x^1 = D_x^4 = \partial_x^h \ , \quad D_y^2 = D_y^3 = \partial_y^h \ , \tag{5.11}$$

or

$$D_x^1 = D_x^{2*} = \partial_x^F \ , \quad D_y^2 = D_y^{3*} = \partial_y^F \ , \tag{5.12}$$

or ∂^B replacing one or both of the ∂^F. Here $\partial^{B*}=\partial^F$, $\partial^{F*}=\partial^B$. Approximations of the form (5.12) could give central approximations to $-\Delta$, but (5.10) with (5.12) is not a central approximation to (5.1), and its truncation error is therefore $O(h)$. Thus we prefer to use (5.11). This we can do only by using staggered grids for U^h and V^h.

The grid we use and the positioning of the discrete variables are shown in Figure 5.1. With this positioning we can indeed approximate (5.1) by

$$\partial_x^h U^h + \partial_y^h V^h = F_1^h \ , \text{ at cell centers } (1) \tag{5.13a}$$

$$\partial_y^h U^h - \partial_x^h V^h = F_2^h \ , \text{ at interior vertices } (2) \ , \tag{5.13b}$$

and the symbol is that of the 5-point Laplacian $-\Delta^h = -\partial_{xx}^h - \partial_{yy}^h$, namely

$$\hat{L}_h(\theta_1,\theta_2) = 4 \sin^2 \frac{\theta_1}{2} + 4 \sin^2 \frac{\theta_2}{2} \ .$$

This symbol vanishes only for $\theta_1 \equiv \theta_2 \equiv 0 \pmod{2\pi}$. Thus (5.13) is an elliptic (even R-elliptic) difference system.

For simplicity we consider here only domains with boundary along grid lines. It is then simple to discretize the boundary condition (5.3). On each boundary link (the heavy lines in Figure 5.1) the variable $(u,v)_n$ is already defined

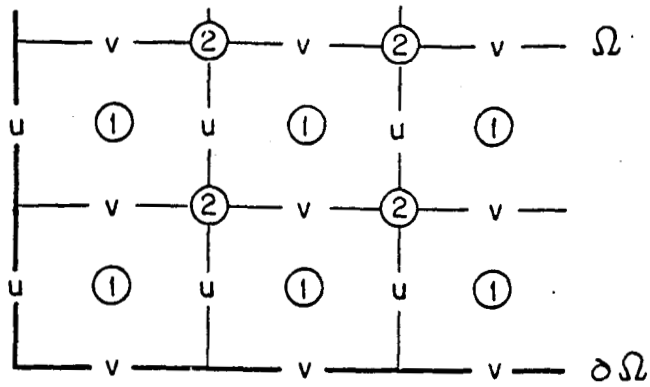


FIGURE 5.1 Discretization of Cauchy-Riemann Equations.
A typical part of the grid is shown. The discrete unknown functions U^h and V^h and their computed approximations u^h and v^h (u and v in the figure) are defined at the centers of vertical and horizontal links, respectively. The first equation (5.13a) is centered at cell centers, where its right-hand side, F_1^h is defined and where (1) is shown in the figure. The second equation (5.13b) is centered, and F_2^h is defined, at the grid vertices, as shown by (2) in the figure.

---

at the center of the link, so (5.3) is discretized to

$$(U^h, V^h)_n = G^h \quad \text{at midpoints of boundary links.} \qquad (5.15)$$

Summing (5.13a) over all the cells of our domain we get the compatibility condition

$$\sum_{\text{cell centers}} F_1^h(x,y) = \sum_{\text{boundary midpoints}} G^h(x,y) \qquad (5.16)$$

which is the discrete analog of (5.4).

Theorem. *If* (5.16) *holds, then the discrete Cauchy-Riemann equations* (5.13) *with the boundary conditions* (5.15) *have a unique solution.*

Indeed, the total number of equations (5.13), (5.15) equals the total number of cells and vertices in the grid. The number of discrete unknowns is the number of links. Hence, by a well-known formula of Euler, there is one more equation than unknowns. But the equations are dependent, as we saw in

constructing the compatibility condition (5.16). Hence, if (5.16) holds, we can remove an equation and have the same number of equations as unknowns. It is therefore enough to prove the theorem for the homogeneous case $F_1^h \equiv 0$, $F_2^h \equiv 0$, $G^h \equiv 0$. In this case (5.13a) implies the existence of a discrete "stream function" ψ^h, defined at the vertices of the grid, such that $U^h = \partial_y^h \psi^h$, $V^h = -\partial_x^h \psi^h$. The homogeneous (5.13b) yields $\Delta^h \psi^h \equiv 0$, and the homogeneous (5.15) implies that ψ^h along the boundary vertices is constant. Hence, by the maximum principle, ψ^h is constant everywhere. Thus, in the homogeneous case $U^h \equiv 0$ and $V^h \equiv 0$, which is what we had to show.

5.3 DGS relaxation and its smoothing rate

Most relaxation schemes are based on one-to-one correspondence between equations and unknowns: The basic relaxation step is to satisfy (or over-satisfy, or under-satisfy) one of the discrete equations by changing the corresponding unknown (or satisfy a group of equations by changing the corresponding group of unknowns). Such one-to-one correspondence is not always natural. In our case, it is clear already in the differential equations (5.1) that it would be unnatural to regard (5.1a), say, as the equation corresponding to the unknown U, and (5.1b) as the one corresponding to V. The entire system corresponds to (U,V). In the difference equations it would be impossible to have even a one-to-one correspondence between pairs of equations and pairs of unknowns, since the number of unknowns is one less than the number of equations. (If the discrete equations (5.12) were used, it would be possible to employ the CSOR scheme introduced in Sec. 4.1, with any $0 < \omega < 1$.)

We will therefore use "distributive relaxation", i.e., a relaxation scheme that satisfies each discrete equation in its turn by distributing changes to several unknowns, in a natural manner.

To derive a natural distributive scheme we note that neither (5.13a) nor (5.13b) are elliptic equations by themselves. It is their combination together which is elliptic.

Hence, in relaxing (5.13a), for example, we should take (5.13b) into account. The simplest way to do it is to relax (5.13a) in such a way that equations (5.13b) are not "damaged", i.e., in a way which preserves the residuals of (5.13b). We do this by simultaneously changing four unknowns, in the following way:

Let (u^h,v^h) be the current approximation to (U^h,V^h). Let (x,y) be the cell center where we next wish to relax (5.13a), and let

$$r_1^h = F_1^h - \partial_x^h u^h - \partial_y^h v^h \tag{5.17}$$

be the "dynamic residual" at (x,y). That is, r is the residual at (x,y) just before relaxing there. The relaxation step of (5.13a) at (x,y) is made up of the following four changes:

$$\begin{aligned} u^h(x+\tfrac{h}{2},y) &\leftarrow u^h(x+\tfrac{h}{2},y) + \delta \\ u^h(x-\tfrac{h}{2},y) &\leftarrow u^h(x-\tfrac{h}{2},y) - \delta \\ v^h(x,y+\tfrac{h}{2}) &\leftarrow v^h(x,y+\tfrac{h}{2}) + \delta \\ v^h(x,y-\tfrac{h}{2}) &\leftarrow v^h(x,y-\tfrac{h}{2}) - \delta \end{aligned} \tag{5.18}$$

where

$$\delta = \tfrac{1}{4} h r^h \ . \tag{5.19}$$

It is easy to check that the distribution of changes (5.18) is such that the residuals

$$r_2^h = F_2^h - \partial_y^h u^h + \partial_x^h v^h \tag{5.20}$$

at all neighboring vertices are not changed, whatever the value of δ. The choice of δ (5.19) is made so that after the changes the residual $r_1^h(x,y)$ will vanish. This is in the manner of the Gauss-Seidel relaxation, where old values are replaced by new values so as to satisfy one difference equation. Such schemes may therefore be called *Distributive Gauss Seidel* (DGS) schemes. In case k of the four values changed in (5.18) are boundary values ($k=1$ near boundaries, except near corners), then no such change should be introduced in those values, and (5.19) is replaced by

$$\delta = \tfrac{1}{4-k} h r_1^h \ . \tag{5.21}$$

The relaxation of (5.13b) is made in a similar manner. If (x,y) is the vertex to be relaxed, the relaxation step will include the changes

$$\begin{aligned} u^h(x,y+\tfrac{h}{2}) &\leftarrow u^h(x,y+\tfrac{h}{2}) + \delta \\ u^h(x,y-\tfrac{h}{2}) &\leftarrow u^h(x,y-\tfrac{h}{2}) - \delta \\ v^h(x+\tfrac{h}{2},y) &\leftarrow v^h(x+\tfrac{h}{2},y) - \delta \\ v^h(x-\tfrac{h}{2},y) &\leftarrow v^h(x-\tfrac{h}{2},y) + \delta \quad , \end{aligned} \tag{5.22}$$

where

$$\delta = \tfrac{1}{4}\, h\, r_2^h \quad . \tag{5.23}$$

The distribution (5.22) is such that the residuals r_1^h will be preserved, and δ in (5.23) is such that equation (5.13b) at (x,y) will be satisfied by the changed variables.

The above relaxation steps can be taken in various orders. In our programs, each complete relaxation sweep comprised of two passes: The first pass relaxes equation (5.13a) by (5.18-19), letting (x,y) pass over all cell centers in, say, lexicographic order. The second pass scans all the grid vertices, relaxing (5.13b) by (5.22-23).

Remark. In terms of the discrete "stream function" ψ^h (see Section 5.2) the second pass of this relaxation reduces to the familiar point-by-point Gauss-Seidel relaxation. The first pass may be viewed as a similar relaxation for the discrete "potential function" φ^h, defined by $U^h = \partial_x^h \varphi^h + f^h$, $V^h = \partial_y^h \varphi^h$, $F_2^h = \partial_y^h f^h$.

The smoothing factor can most conveniently be calculated in terms of the residual functions (r_1, r_2). For the Fourier component $\exp(i\theta_1 x/h + i\theta_2 y/h)$, let A be the amplitude of hr_1^h before the first pass, $\bar{A}$ its amplitude after the pass, $\tilde{A}$ the amplitude of the dynamic hr_1^h residuals, and B the amplitude of δ. It is clear from (5.19) that $B = \tilde{A}/4$, and from (5.18)

$$\begin{aligned} \tilde{A} &= A + B\, e^{-i\theta_1} + B\, e^{-i\theta_2} \\ \bar{A} &= B\, e^{i\theta_1} + B\, e^{i\theta_2} \quad . \end{aligned}$$

Hence

$$B = A/(4 - e^{-i\theta_1} - e^{-i\theta_2}) ,$$

and the amplification factor of r_1^h in the first pass is

$$\lambda(\underline{\theta}) = \frac{\bar{A}}{A} = \frac{e^{i\theta_1} + e^{i\theta_2}}{4 - e^{-i\theta_1} - e^{-i\theta_2}} . \tag{5.24}$$

The residuals r_2^h are not changed by the first pass. Similarly, in the second pass the Fourier components of r_2^h are amplified by $\lambda(\underline{\theta})$, while r_1^h remains unchanged. In a complete sweep the amplitude of the vector (r_1^h, r_2^h) is therefore amplified by the "amplification matrix"

$$\begin{pmatrix} \lambda(\underline{\theta}) & 0 \\ 0 & \lambda(\underline{\theta}) \end{pmatrix} . \tag{5.25}$$

Hence the smoothing factor is

$$\bar{\mu} = \max_{|\underline{\theta}| \geq \frac{\pi}{2}} |\lambda(\underline{\theta})| = .5 \tag{5.26}$$

Unsurprisingly, this smoothing factor is the same as in Gauss-Seidel relaxation for the 5-point Laplacian. The convergence rate of relaxation is also essentially the same as for Poisson problems, as was confirmed by numerical experiments.

5.4. Multi-grid procedures

Assume now we have a sequence of grids (levels) with mesh-sizes $h_1, \ldots, h_M$, where $h_{k+1} = \frac{1}{2}h_k$. The relative position of the different grids is shown in Figure 5.2. Instead of F_1^h, F_2^h, G^h, U^h, V^h, u^h, v^h, r_1^h and r_2^h used above, the discrete functions on the k-th level will be denoted by F_1^k, F_2^k, G^k, U^k, V^k, u^k, v^k, r_1^k and r_2^k, respectively. The multi-grid algorithm we use is the accommodative Cycle C algorithm (see Section 2.6 above, or Section 4 in [B3]). For relaxation we employ the DGS sweeps described in Section 5.3 above.

The coarse-to-fine interpolation can be of first order, since this is the highest order of derivatives in the Cauchy-Riemann operator. An obvious way of doing such an interpolation (see Figure 5.2) is

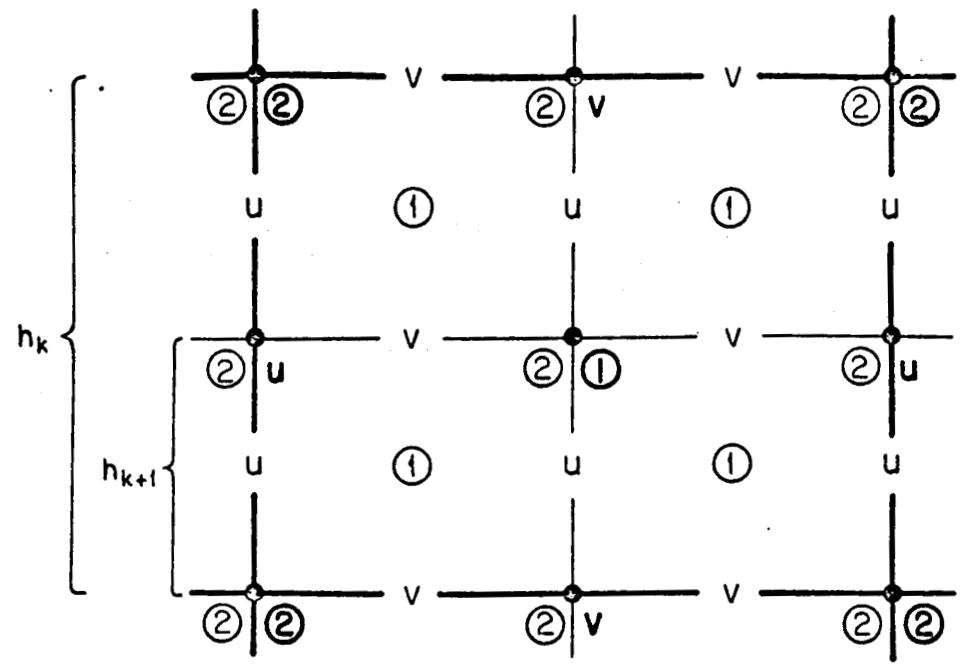


FIGURE 5.2 A coarse-grid cell divided into fine-grid cells. Same notations as in Figure 5.1, with heavier-type being used for the coarse-grid and lighter-type for the fine-grid.

$$I_k^{k+1}\, u^k(x,y) = \begin{cases} u^k(x,y+\tfrac{1}{2}h_{k+1}) \quad \text{or} \quad u(x,y-\tfrac{1}{2}h_{k+1}) \quad \text{if } x \\ \qquad\qquad \text{is on a coarse-grid line} \\ \tfrac{1}{2}[I_k^{k+1}u^k(x+h_{k+1},y) + I_k^{k+1}\, u^k(x-h_{k+1},y)] \\ \qquad\qquad \text{otherwise} \end{cases} \tag{5.27}$$

and similarly for $I_k^{k+1}\, v^k$. One can of course use linear interpolations instead.

The Cauchy-Riemann problem is linear. We can therefore make coarse-grid corrections either by the Correction Scheme or the Full-Approximation Scheme (FAS). In the latter case we have to define the fine-to-coarse transfer of solution $(I_{k+1}^k\, u^{k+1}, I_{k+1}^k\, v^{k+1})$. We use the following averaging (see the coarse-grid and fine-grid positions of u and v in Figure 5.2) :

$$\begin{aligned} I_{k+1}^k\, u^{k+1}(x,y) &= \mu_y^{k+1}\, u^{k+1}(x,y) \\ &\equiv \tfrac{1}{2}\,[u^{k+1}(x,y+\tfrac{1}{2}h_{k+1}) + u^{k+1}(x,y-\tfrac{1}{2}h_{k+1})] \end{aligned} \tag{5.28a}$$

$$I_{k+1}^{k}\ v^{k+1}(x,y) = \mu_x^{k+1}\ v^{k+1}(x,y) \qquad (5.28b)$$

$$\equiv \frac{1}{2}\{v^{k+1}(x+\tfrac{1}{2}h_{k+1},y)+v^{k+1}(x-\tfrac{1}{2}h_{k+1},y)\}\ .$$

The fine-to-coarse transfer of residuals $I_{k+1}^{k} r_1^{k+1}$ (residuals of the first equation, at cell centers) is also done by averaging:

$$I_{k+1}^{k}\ r_1^{k+1} = \mu_x^{k+1}\ \mu_y^{k+1}\ r_1^{k+1}\ . \qquad (5.29)$$

(See the coarse-grid and fine-grid positions of equations 1 in Figure 5.2.) When the Correction Scheme is used, (5.29) serves as the right-hand side of equation (5.13a) on the coarser level h_k. In calculating (5.29) using (5.17), observe that some terms are cancelled and some of the additions can be made only once for two neighboring coarse-grid cells. It is interesting to note that when FAS is used it is not necessary to calculate (5.29). Transferring u^{k+1} and v^{k+1} by (5.28) and residuals by (5.29), it is easy to see that the FAS coarse-grid equation will read

$$\partial_x^k\ u^k + \partial_y^k\ u^k = \mu_x^{k+1}\ \mu_y^{k+1}\ F_1^{k+1}\ . \qquad (5.30)$$

Thus, the coarse-grid equation in this case is not affected at all by the fine-grid solution. If we let $F_1^k = \mu_x^k\ \mu_y^k\ F_1^{k+1}$ we find that (5.30) is actually identical with (5.13a) for the k-th level. In other words, the relative truncation error in (5.13a) vanishes.

Another nice feature of (5.30) is that if the compatibility condition (5.16) is satisfied on the fine grid, it will automatically be satisfied in the coarse grid problem (upto round-off errors, of course).

The residuals of (5.13b) can be transferred to the coarse grid by "injection" :

$$I_k^{k+1}\ r_2^{k+1}(x,y) = r_2^{k+1}(x,y)\ , \qquad (5.31)$$

since any coarse-grid center of that equation (any coarse-grid vertex) coincides with a fine-grid center of the equation (a fine-grid vertex).

We have made experiments with the Cycle C algorithm only. For FMG algorithms, a higher-order interpolation routine should be added. This interpolation in the present case needs to be at least quadratic (order 3).

5.5 Multi-grid results

Numerical experiments with this algorithm are reported in [D1]. They show, unsurprisingly, exactly the same convergence as in multi-grid solutions for Poisson problems; namely, a convergence factor of about .55 per RWU (relaxation work unit). Indeed, the entire procedure can be described as a multi-grid process for two Poisson problem. (One Poisson problem in terms of the stream function for the case $F_1 \equiv 0$, the other in terms of the potential function for the case $F_2 \equiv 0$.) Hence although experiments were conducted with Cycle C algorithm only, it can be safely predicted that the Fixed FMG algorithm (Section 2.2) will solve the problem to within the truncation errors (and even far below, when τ-extrapolation is employed), in 5.3 RWU.

The number of operations in such a CS algorithm, taking into account the relaxation sweeps and all the coarse-to-fine and fine-to-coarse transfers, is about $61n$, where n is the number of unknowns in the finest grid. Almost all these operations are either additions or shifts (i.e., multiplications by an integer power of 2); less than $3.5n$ of them are real multiplications. In fact, these $3.5n$ multiplications (needed in the quadratic interpolations) can be replaced by $4n$ additions plus $2n$ shifts.

There is a faster way for solving the discrete Cauchy-Riemann equations (5.13): Subtracting from U^h a function U_0^h which satisfies $\partial_x^h U_0^h \equiv F_1^h$, we get a new system in which $F_1^h \equiv 0$. The problem can then be rewritten as a Poisson problem for the discrete stream function ψ^h. Solving that Poisson problem by a similar Full Multi-Grid algorithm, together with the operations of subtracting U_0^h and constructing U^h and V^h would require about $23.5n$ operations (additions and shifts only. Cf. [B3]). The main purpose of this chapter, however, was to study methods for solving elliptic *systems*. The techniques developed for the present simple system are applicable to much more complicated ones.

6. STEADY-STATE STOKES EQUATIONS

6.1 The differential problem

As a prelude to the treatment of the full Navier-Stokes equations, we consider now the steady-state Stokes equations in a d-dimensional domain

$$\underline{\nabla}\cdot\underline{U} = F_0 \tag{6.1a}$$

$$-\Delta\underline{U} + \underline{\nabla}P = \underline{F} \quad , \tag{6.1b}$$

where $\underline{U} = (U_1,\ldots,U_d)$ represents the velocity of a fluid and P represents the pressure, $\underline{\nabla} = (\partial_1,\ldots,\partial_d)$ is the gradient operator, $\Delta = \partial_1^2 + \ldots + \partial_d^2$ is the Laplace operator, and F_0 and $\underline{F} = (F_1,\ldots,F_d)$ are given forcing functions. (6.1) are the equations of "creeping" flows (vanishing Reynolds number). (6.1a) is the "continuity equation" (usually with vanishing source term: $F_0 \equiv 0$), and (6.1b) is the vector of d momentum equations.

The matrix-operator form of (6.1) is

$$L \begin{pmatrix} P \\ U_1 \\ \vdots \\ U_d \end{pmatrix} \equiv \begin{pmatrix} 0 & \partial_1 & \cdots & \partial_d \\ \partial_1 & -\Delta & & \bigcirc \\ \vdots & & \ddots & \\ \partial_d & \bigcirc & & -\Delta \end{pmatrix} \begin{pmatrix} P \\ U_1 \\ \vdots \\ U_d \end{pmatrix} = \begin{pmatrix} F_0 \\ F_1 \\ \vdots \\ F_d \end{pmatrix} \tag{6.2}$$

and the operator determinant is

$$\det L = (-\Delta)^d \quad . \tag{6.3}$$

Hence (6.1) is a 2d-order elliptic system and will require d boundary conditions. These are usually given by specifying the velocity on the boundary

$$\underline{U}(x) = \underline{G}(x) \quad , \qquad (x \in \partial\Omega) \quad , \tag{6.4}$$

where $\underline{G} = (G_1,\ldots,G_d)$.

Equations (6.1) with the boundary conditions (6.4) constitute a well-posed problem, provided the compatibility condition

$$\int_{\Omega} F_0 \, d\underline{x} = \int_{\partial\Omega} \underline{G}\cdot d\underline{\sigma} \tag{6.5}$$

is satisfied, where $d\underline{\sigma}$ is the boundary element multiplying an outward normal unit vector.

6.2 Finite-difference equations

By arguments similar to those in Section 5.2, we find it best to discretize (6.1) on a staggered grid. Such a grid, in the two-dimensional case, is shown in Fig. 6.1. In the general d-dimensional case, the grid planes define cells, each cell with $2d$ faces. The discrete velocity U_j^h and its computed approximations u_j^h are defined at centers of j-faces, i.e., faces perpendicular to the j-th coordinate. The discrete

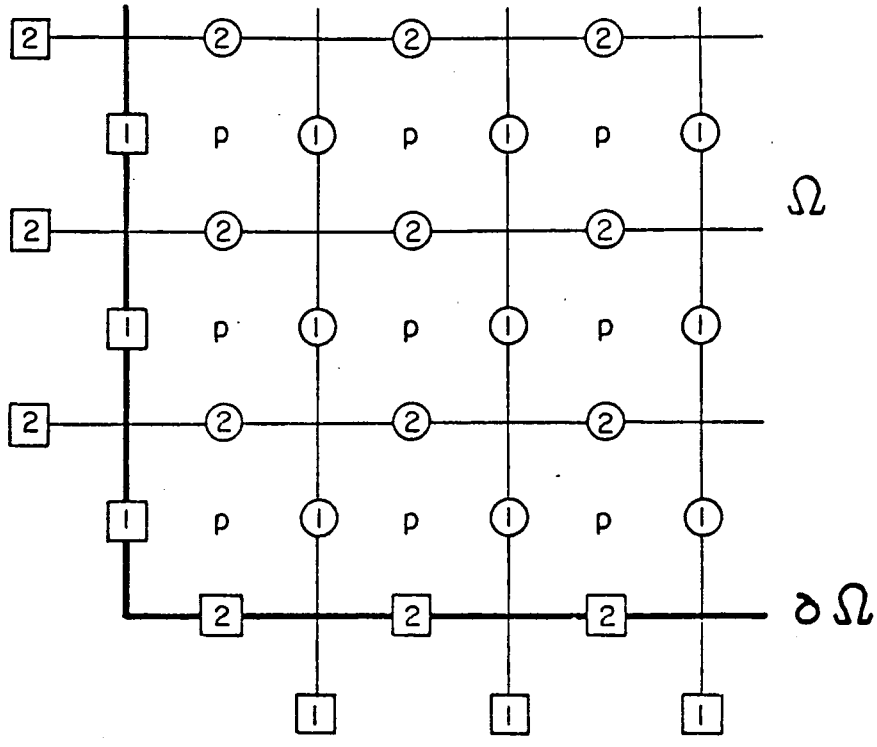


FIGURE 6.1 Discretization of two-dimensional Stokes Equations
A typical part of the grid is shown. The discrete pressure p^h is defined at cell centers (p). The discrete velocity u_1^h is defined at centers of vertical links ((1) = interior centers; [1] = boundary and exterior centers), and u_2^h is defined at centers of horizontal links ((2) and [2]). The discrete continuity equations are centered at cell centers (p). The j-th momentum equation is centered at interior values of u_j^h ((j)). The exterior values of u_1^h and u_2^h (at [1] and [2], respectively, but not on the boundary) are fictitious.

pressure P^h and its computed approximation p^h are located at cell centers. The discrete approximation to (6.1) can then be written (with the notation of Sec. 3.2) as

$$\sum_{j=1}^{d} \partial_j^h U_j^h = F_0^h \qquad \text{at cell centers} \tag{6.6a}$$

$$-\Delta^h U_j^h + \partial_j^h P = F_j^h \qquad \text{at centers of } j\text{-faces}, \quad (j = 1,\ldots,d) \tag{6.6b}$$

where the discrete approximation Δ^h to Laplace operator is the usual (2d+1)-point approximation $\sum_1^d (\partial_j^h)^2$. For a point $\underline{x}$ near a boundary, however, $\Delta^h U_j^h(\underline{x})$ may involve an exterior value $U_j^h(\underline{x}^e)$. This value is defined by quadratic extrapolation from $U_j^h(\underline{y})$, $U_j^h(\underline{x})$ and $U_j^h(\underline{x}^b) = G_j^h(\underline{x}^b)$, where $\underline{x}^b$ is a boundary point on the segment $(\underline{x},\underline{x}^e)$.* This definition is used to eliminate the exterior value from $\Delta^h U_j^h(\underline{x})$, so that the discrete Laplacian is modified and includes a boundary value of U_j^h.

The matrix operator of (6.6) is

$$L^h = \begin{pmatrix} 0 & \partial_1^h & \cdots & \partial_d^h \\ \partial_1^h & -\Delta^h & & \bigcirc \\ \vdots & & \ddots & \\ \partial_d^h & \bigcirc & & -\Delta^h \end{pmatrix}, \tag{6.7}$$

hence $\det L^h = (-\Delta^h)^d$ and its symbol is

$$\hat{L}^h(\underline{\theta}) = \tilde{L}^h(\underline{\theta}) = \left\{ \sum_1^d (2 \sin \frac{\theta_j}{2})^2 \right\}^d, \tag{6.8}$$

which is positive for $0 < |\theta| \leq \pi$. The difference system (6.6) is therefore R-elliptic.

The boundary condition (6.4) is approximated by the way we treat boundary and exterior values of U_j^h. For simplicity we

---

*Note that different interior points may be adjacent to the same exterior point $\underline{x}^e$. The extrapolated exterior value depends on $\underline{x}$ and $\underline{x}^b$, hence slightly different exterior values may correspond to the same exterior point.

consider the case of domains whose boundary is contained in grid lines (or grid planes). In this case the velocity normal to the boundary is conveniently defined at the center of boundary faces, and the discrete analog to (6.5) is naturally written as

$$\sum_{\underline{x}} F_0^h(\underline{x}) = \sum_{\underline{y}} G_n^h(\underline{y}) \quad , \tag{6.9}$$

where $\underline{x}$ runs over all cell centers, $\underline{y}$ runs over all centers of boundary faces, and $G_n^h(\underline{y})$ is the (given) normal velocity at $\underline{y}$.

Theorem. *The discrete Stokes equations (6.6), with exterior and boundary values determined by the boundary conditions as above, have a unique solution if and only if (6.9) is satisfied.*

The proof is simple. The number of equations is the same as the number of unknowns, since for each interior $U_j^h(\underline{x})$ there corresponds an equation (6.6b) at $\underline{x}$, and for each unknown $P^h(\underline{y})$ there corresponds an equation (6.6a) at $\underline{y}$. The pressure values P^h are determined only upto an additive constant, but, on the other hand, the equations are dependent; summing (6.6a) over all cell centers we get (6.9). That is to say, if (6.9) is not satisfied we get a contradiction. If (6.9) is satisfied, we get a dependence of equations, corresponding to the arbitrary constant in P^h . Hence, it is enough to show that in the homogeneous case ($\underline{F}^h \equiv 0$, $\underline{G}^h \equiv 0$) , the only solution is the trivial one ($\underline{U}^h \equiv 0$, $P^h \equiv$ constant). Indeed, if $\underline{F}^h \equiv 0$ it is easy to see from (6.6b) that

$$\begin{aligned} 0 &= \sum_{j=1}^{d} \sum_1 [-\Delta^h U_j^h(\underline{x}) + \partial_j^h P^h(\underline{x})]\ U_j^h(\underline{x}) \\ &= \sum_{j=1}^{d} h^{-2} \sum_2 [U_j^h(\underline{x}) - U_j^h(\underline{y})]^2 \\ &\quad + \sum_{j=1}^{d} h^{-2} \sum_3 [U_j^h(\underline{x}) - U_j^h(\underline{z})]\ U_j^h(\underline{x}) \\ &\quad + \sum_4 P^h(\underline{x}) \sum_{j=1}^{d} \partial_j^h U_j^h(\underline{x}) \ , \end{aligned}$$

where the point $\underline{x}$ in Σ_1 runs over all interior positions of U_j^h (points (j) in Fig. 6.1); the pair $\{\underline{x},\underline{y}\}$ in Σ_2

runs over all pairs of neighboring interior positions of U_j^h; the pair $\{\underline{x},\underline{z}\}$ in Σ_3 runs over all pairs of neighboring positions U_j^h, with $\underline{x}$ being an interior position ((j) in Fig. 6.1), and $\underline{z}$ being a boundary or exterior position ([j] in Fig. 6.1); and $\underline{x}$ in Σ_4 runs over all cell centers (p in Fig. 6.1). The term with Σ_4 vanishes by (6.6a), since $F_0^h \equiv 0$. In the Σ_3 term, by the way exterior values are defined, we get (for $G \equiv 0$) $U_j^h(z) = 2\,U_j^h(\underline{x}) - \frac{1}{3}\,U_j^h(\underline{y})$, where $\underline{y}$ is the interior neighbor of $\underline{x}$ opposite $\underline{z}$. Hence,

$$\sum_{j=1}^{d}\left\{\sum_2'[U_j^h(\underline{x}) - U_j^h(\underline{y})]^2 + \sum_3 4\,U_j^h(\underline{x})^2 + U_j^h(\underline{y})^2 - \frac{7}{3}\,U_j^h(\underline{x})\,U_j^h(\underline{y})\right\} = 0$$

where $\sum_2'$ runs as $\sum_2$ except for terms added to $\sum_3$. This form is positive definite, hence $U_j^h \equiv 0$. By (6.6b) $P^h \equiv$ const.

6.3 Distributive relaxation

The j-th momentum equation (6.6b) is elliptic in U_j^h. We will therefore smooth the residuals of that equation by relaxing it in the following natural way: For a fixed j we scan in some order all the interior points $\underline{x}$ where U_j^h is defined. At each such point $\underline{x}$ we change $U_j^h(\underline{x})$ so as to satisfy the j-th momentum equation centered at $\underline{x}$.

Having done such a sweep for each $j = 1,\ldots,d$, we now need to smooth the error in the continuity equation (6.6a). The remaining variable left to be relaxed is P^h, which seems indeed to correspond "geographically" to the continuity equation, i.e., P^h is defined where (6.6a) is centered. But P^h does not even appear in (6.6a), so by itself it cannot be used to relax that equation. Here we recall our lesson from the Cauchy-Riemann equations (Section 5.3): Equation (6.6a) is not elliptic, it is only a part of an elliptic system. The way to relax it is therefore by a distributive relaxation designed so as to keep unchanged the residuals of the other equations in the system. It is done as follows:

Let $(p^h,u_1^h,\ldots,u_d^h)$ be the current approximation to $(P^h,U_1^h,\ldots,U_d^h)$. We scan the cell centers in some preassigned order. Let $\underline{x}$ be the current cell center and let

$$r_0^h(\underline{x}) = F_0^h(\underline{x}) - \sum_{j=1}^{d} \partial_j^h u_j^h(\underline{x}) \qquad (6.10)$$

be the "dynamic residual" at $\underline{x}$; i.e., the residual at $\underline{x}$ just before relaxing there. The relaxation step at $\underline{x}$ is made up of the following 4d+1 changes (see Fig. 6.2(a)):

$$u_j^h(\underline{x} + \tfrac{1}{2}\underline{h}_j) \leftarrow u_j^h(\underline{x} + \tfrac{1}{2}\underline{h}_j) + \delta \ , \quad (j = 1,\ldots,d) \ , \qquad (6.11a)$$

$$u_j^h(\underline{x} - \tfrac{1}{2}\underline{h}_j) \leftarrow u_j^h(\underline{x} - \tfrac{1}{2}\underline{h}_j) - \delta \ , \quad (j = 1,\ldots,d) \ , \qquad (6.11b)$$

$$p^h(\underline{x}) \leftarrow p^h(\underline{x}) + \frac{2d}{h}\,\delta \quad , \qquad (6.11c)$$

$$p^h(\underline{x} + \underline{h}_j) \leftarrow p^h(\underline{x} + \underline{h}_j) - \frac{1}{h}\,\delta \ , \quad (j = 1,\ldots,d) \ , \qquad (6.11d)$$

$$p^h(\underline{x} - \underline{h}_j) \leftarrow p^h(\underline{x} - \underline{h}_j) - \frac{1}{h}\,\delta \ , \quad (j = 1,\ldots,d) \ , \qquad (6.11e)$$

where

$$\delta = \frac{h}{2d}\, r_0^h(\underline{x}) \quad , \qquad (6.12)$$

and where $\underline{h}_j$ is h times the unit vector in the x_j direction. Like (5.18-19) above, changes (6.11a,b), (6.12) are such that, *after* changing, $r_0^h(\underline{x})$ vanishes. The pressure changes (6.11c,d,e) are such that the momentum-equations residuals

$$r_j^h = F_j^h + \Delta^h u_j^h - \partial_j^h p^h \quad , \qquad (6.13)$$

at all points remain unchanged.

Indeed, another way of writing the relaxation step at the cell center $\underline{x}$ is through the characteristic function of that cell, which we denote by $\chi_{\underline{x}}^h$; that is, $\chi_{\underline{x}}^h$ is a function defined at cell centers, with $\chi_{\underline{x}}^h(\underline{\xi}) = 0$ except for $\chi_{\underline{x}}^h(\underline{x}) = 1$. Changes (6.11) can be written as

$$\begin{aligned} u_j^h &\leftarrow u_j^h - \delta h \partial_j^h \chi_{\underline{x}}^h \qquad (j = 1,\ldots,d) \\ p^h &\leftarrow p^h - \delta h \Delta^h \chi_{\underline{x}}^h \quad . \end{aligned} \qquad (6.14)$$

Substituting these changes into (6.13) we immediately get

$$\begin{aligned} r_j^h &\leftarrow r_j^h - \delta h \Delta^h \partial_j^h \chi_{\underline{x}}^h + \delta h \partial_j^h \Delta^h \chi_{\underline{x}}^h \\ &= r_j^h \ , \qquad (j = 1,\ldots,d) \ . \end{aligned}$$

Near the boundary it is not possible to precisely preserve $r_1^h,\ldots,r_d^h$ while relaxing the continuity equation.

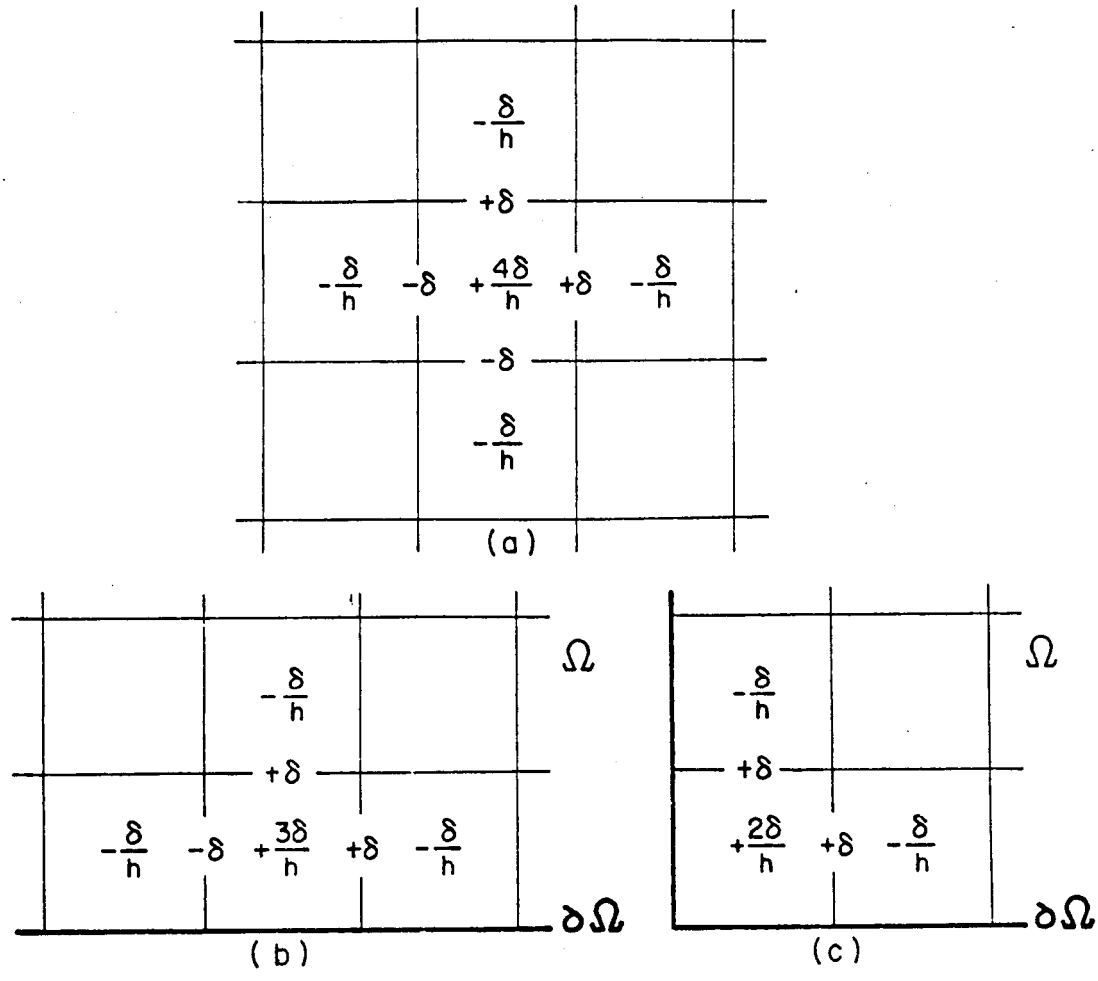


FIGURE 6.2 Continuity-Equation Relaxation Step in 2-Dimensional Stokes Equations.

(a) The cell at the center of the figure is relaxed by 9 simultaneous changes. The amount of change is displayed at the position of the changed variable. (cf. Fig.6.1.)
$\delta = h\, r_0^h(\underline{x})/4$, where $r_0^h(\underline{x})$ is the dynamic residual at the relaxed cell.

(b) Configuration of changes in a boundary cell.
$\delta = h\, r_0^h(\underline{x})/3$.

(c) Configuration of changes in a corner cell.
$\delta = h\, r_0^h(\underline{x})/2$.

Neither is it necessary. It is enough to relax r_0^h so that the changes introduced to $r_1^h,\ldots,r_d^h$ do not cause later (when the momentum equations are relaxed) significant "feed-back", i.e., too large changes back in r_0^h. Near the boundary feed-back changes are partly "absorbed" by the boundary conditions, and therefore such "small feed-back" schemes are easy to design. If for example we use the scheme shown in Figs. 6.2(b) or 6.2(c), it is easy to calculate and show that feed-back changes in r_0^h are small enough. That is, the sum of all feed-back changes in r_0^h is zero (hence, smooth errors contribute little to the feed-back), while the sum of their absolute values is only a small fraction of the relaxed quantity. Moreover, the signs of the feed-back residual changes is opposite to the direct changes in r_0^h caused by the step shown in Figs. 6.2(b) or 6.2(c).

The smoothing factor is most easily calculated by the amplification matrix of the residuals $r_0^h,r_1^h,\ldots,r_d^h$. For the Fourier component $\exp(i\,\underline{\theta}\cdot\underline{x}/h) = \exp(i\Sigma\theta_j x_j/h)$, let $(A_0,A_1,\ldots,A_d)$ be the amplitudes of $(r_0^h,r_1^h,\ldots,r_d^h)$ before the relaxation sweep, and let $(\bar{A}_0,\bar{A}_1,\ldots,\bar{A}_d)$ be the corresponding amplitudes after the sweep. The sweep is made of d+1 passes. In the j-th pass $(j = 1,\ldots,d)$, relaxing the j-th momemtum equation, A_j is multiplied by $\lambda(\theta)$, while other momemtum amplitudes A_k $(1 \leq k \leq d,\ k \neq j)$ remain unchanged.

$$\lambda(\underline{\theta}) = \frac{\beta}{2d - \bar{\beta}}, \quad \text{where} \quad \beta(\underline{\theta}) = \sum_{j=1}^{d} e^{i\theta_j}, \tag{6.15}$$

is the Gauss-Seidel amplification factor for the (2d+1)-point Poisson equation (cf. e.g., Sec. 3.1 in [B3]). The j-th pass does change A_0, adding to it some multiple of A_j. In the last pass, relaxing the continuity equation, $A_1,\ldots,A_d$ remain unchanged, and A_0 is multiplied by $\lambda(\theta)$ (cf. Sec. 5.3 above). Hence

$$\begin{pmatrix} \bar{A}_0 \\ \bar{A}_1 \\ \vdots \\ \bar{A}_d \end{pmatrix} = \begin{pmatrix} \lambda(\underline{\theta}) & \lambda_{01} & \cdots & \lambda_{0d} \\ 0 & \lambda(\underline{\theta}) & & \bigcirc \\ \vdots & & \ddots & \\ 0 & \bigcirc & & \lambda(\underline{\theta}) \end{pmatrix} \begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_d \end{pmatrix}. \tag{6.16}$$

The largest eigenvalue of this (triangular) amplification matrix is $\lambda(\underline{\theta})$, hence the smoothing factor is

$$\bar{\mu} = \max_{\frac{\pi}{2} \le |\theta| \le \pi} |\lambda(\underline{\theta})| = \begin{cases} .500 & \text{if } d = 2 \\ .567 & \text{if } d = 3 \end{cases} , \qquad (6.17)$$

the same smoothing factor as in Gauss-Seidel relaxation for the standard Poisson equation.

6.4 Multi-grid procedures

For multi-grid processing of Stokes equations we use a sequence of grids (levels) with sizes $h_1, \ldots, h_M$, where $h_{k+1} = \frac{1}{2} h_k$, and where the grid lines (or grid planes) of level k are every other grid line (plane) of level $k+1$. Hence, each cell of level k is the union of 2^d cells of level $k+1$. In two dimensions (d=2) the configuration is shown in Fig. 6.3. Instead of $\underline{F}^h$, $\underline{U}^h$, P^h , $\underline{r}^h$, μ^h and ∂_j^h used in Sections 6.2 and 6.3, the discrete functions and operators on the k-th level are now denoted by $\underline{F}^k$, $\underline{U}^k$, $\underline{P}^k$, C^k , μ^k and ∂_j^k , respectively.

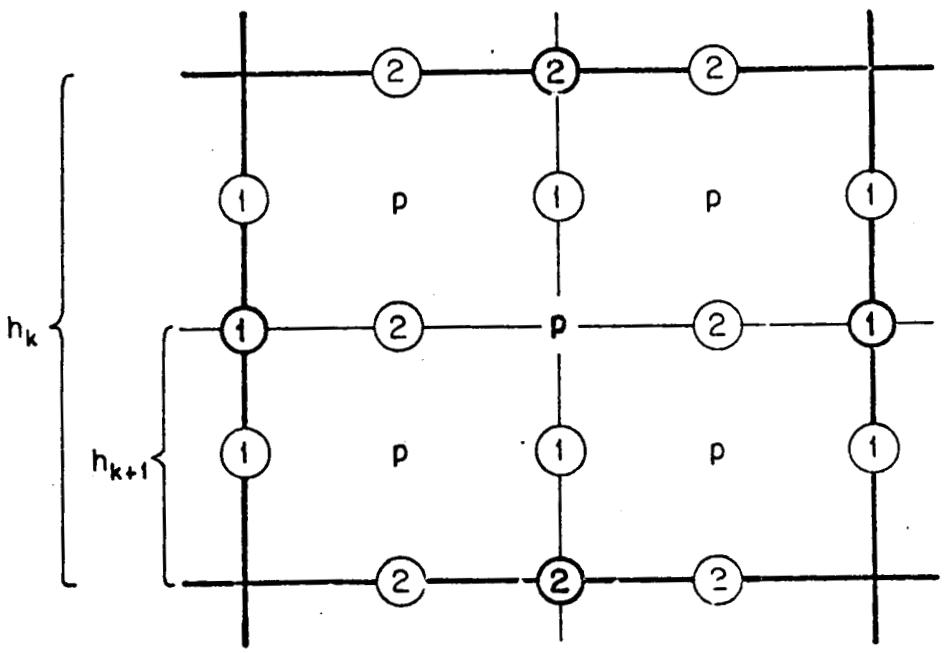


FIGURE 6.3 A coarse-grid cell divided into fine-grid cells.
Same notation as in Fig. 6.1 is used, with heavy type for the coarse grid and light type for the fine grid.

We have solved Stokes equations using both Cycle C and the Full Multi-Grid (FMG) algorithms (see Secs. 2.2 and 2.6). For coarse-grid corrections we used alternatively the Correction Scheme (CS) and the Full-Approximation Scheme (FAS), with identical results. We describe here the procedures in terms of FAS, since CS is not extendable to the nonlinear Navier-Stokes equations.

Coarse-to-fine interpolations. In the FMG algorithm, the first coarse-to-fine interpolation (2.11) has to be of order at least four for the velocities and at least three for the pressure. The design of such interpolations is straightforward, although it turns out somewhat cumbersome near boundaries.

The coarse-to-fine interpolation of corrections (I_k^{k+1} in (2.15)) has to be of orders at least two for the velocities and one for the pressure. We used bilinear (i.e., order two) interpolations for both.

The fine-to-coarse transfers are made by averaging. For the FAS transfer of u_j^{k+1} we can use the same averaging as for the r_j^{k+1} , $(j = 1,\ldots,d)$, which can be either the minimal-operations transfer

$$I_{k+1}^{k}\, r_j^{k+1} = \mu_1^{k+1} \cdots \widehat{\mu_j^{k+1}} \cdots \mu_d^{k+1}\, r_j^{k+1} \quad , \qquad (j = 1,\ldots,d) \tag{6.18}$$

or the full weighting

$$I_{k+1}^{k}\, r_j^{k+1} = \mu_j^{k+1}\, \mu_1^{k+1} \cdots \mu_d^{k+1}\, r_j^{k+1} \quad , \qquad (j = 1,\ldots,d) \tag{6.19}$$

where the hat in (6.18) indicates the term to be skipped in the sequence. The residual-weighting (6.18) is less expensive than (6.19), especially since it requires calculating only one half of the fine-grid residuals. But (6.19) is more reliable in the nonlinear case and near boundaries, since it is "full" i.e., it transfers all the fine-grid residuals, attaching the same total weight to each of them.

The FAS transfer of p^{k+1} can be made with the same weighting as the transfer of the continuity-equation residuals

$$I_{k+1}^{k}\, r_0^{k+1} = \mu_1^{k+1} \cdots \mu_d^{k+1}\, r_0^{k+1} \quad , \tag{6.20}$$

which is both simplest and full. In fact, if the minimal-operations transfer (6.18) is used for the velocities u_j^{k+1}, then (6.20) need not really be calculated: If the FAS continuity equation on level k is written in the form

$$\sum_{j=1}^{d} \partial_j^k U_j^k = f_0^k \qquad (6.21a)$$

(where $f_0^\ell \equiv F_0^\ell$ on the currently finest level ℓ), it is easy to see that (6.20) is equivalent to

$$f_0^k = \mu_1^{k+1} \ldots \mu_d^{k+1} f_0^{k+1} , \qquad (k < \ell) , \qquad (6.21b)$$

which does not depend on $\underline{u}^{k+1}$.

The compatibility condition (6.9) is automatically obtained (upto round-off errors) on all levels provided it holds on the finest one. This results directly from (6.20).

Switching criteria. Since the rate of smoothing of all the relaxation passes is the same, we could base our algorithm (see Sec. 2.2)) on residual norm of the form

$$e_k = \sum_{0}^{d} \alpha_j \| r_j^k \| , \qquad (6.22)$$

where r_j^k are defined in (6.10), (6.13), and the norm is the L_2 norm, say. There was no sensitivity to the choice of $\alpha_j > 0$, and one could for example use $\alpha_0 = 1$, $\alpha_1 = \ldots = \alpha_1 = 0$. Another alternative is to use a fixed algorithm, such as Fig. 2.2.

τ-extrapolations have first failed to yield impressive improvements. Only later we realized the reason: In a staggered grid, in order to employ τ-extrapolation, the coarse-to-fine interpolation (2.12) must be of a higher order.

6.5 Numerical Results

For programming simplicity we confined our experiments to two-dimensional (d = 2) rectangles. (Experiments with many equations [S1] conclusively show that the shape of the domain does not significantly affect the performance of the algorithm.) We first experimented with cycling algorithms (Sec. 2.6). Since the smoothing factor $\bar{\mu} = .50$ is the same

as $\bar{\mu}$ for the five-point Poisson equation, we were surprised when the experiments showed that the asymptotic multi-grid convergence factor per relaxation-work-unit for the Stokes equation was around $\overset{\circ}{\mu} = .65$, as against $\overset{\circ}{\mu} = .55$ for the Poisson equation. For some time we thought that our treatment of the boundary conditions might be responsible for the slower convergence. The one-level analysis (Sec. 4.4) cannot show any difference between Stokes and Poisson equations; it gives $\overset{\circ}{\mu} = \bar{\mu}^{3/4} = .595$ for both. Therefore the two-level analysis had to be used. That analysis indeed gave a very good agreement with the numerical results (see Table 6.1 and more details in [D2]), which finally satisfied us that the program, including the procedures at the boundary, were all right.

TABLE 6.1

Comparison between theoretical and experimental convergence factors for Stokes equations (reproduced from [D1]).

Number of sweeps on finest grid per cycle	δ (See Step F in Sec. 2.2 & Sec. 4.7.2)	$\overset{\circ}{\mu}$	
		Mode analysis	experimental
1	.4	.661	.638
1	.5	.648	.634
2	.3	.680	.695
2	.4	.710	.725
3	.1	.714	.722

The next step was to convert our program from CS to FAS, and then from cycling to Full Multi-Grid (FMG – as in Sec. 2.2). Later, the Stokes Program was generalized to Navier-Stokes. The FMG results for Stokes equations is a special case of the results described in Sec. 7.5 (and much further in [D2]) for the Navier-Stokes equations.

7. STEADY-STATE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

7.1 The differential problem

The steady-state incompressible Navier-Stokes equations in d dimensions are

$$\underline{\nabla}\cdot\underline{U} = F_0 \tag{7.1a}$$

$$Q\underline{U} + \underline{\nabla}P = \underline{F} \tag{7.1b}$$

where $Q = -\Delta + R\sum U_i\partial_i$, R being the Reynolds number. The Stokes system (6.1) is the special case $R = 0$. In the discussion below, especially in the mode analysis (but not in the actual solution process) we treat Q as being independent of $\underline{U}$, i.e., as if some U_i^0 appears in Q instead of U_i. This is equivalent to linearizing the system around some $\underline{U}^0$, and omitting the lowest order term of the linearized equations (the term $R\sum_i(\partial_i U_j^0)U_j$ in the j-th momentum equation. For all Reynolds numbers this term is locally dominated by the second term of QU_j , on any scale h such that $|h\partial_i U_j| < |\underline{U}|$). We can then write the equations in the matrix form

$$L\begin{pmatrix} P \\ U_1 \\ \vdots \\ U_d \end{pmatrix} \equiv \begin{pmatrix} 0 & \partial_1 & \cdots & \partial_d \\ \partial_1 & Q & & \\ \vdots & & \ddots & \\ \partial_d & & & Q \end{pmatrix}\begin{pmatrix} P \\ U_1 \\ \vdots \\ U_d \end{pmatrix} = \begin{pmatrix} F_0 \\ F_1 \\ \vdots \\ F_d \end{pmatrix} , \tag{7.2}$$

so that $\det L = -\Delta\, Q^{d-1}$. Hence (7.1) is again an elliptic system of order $2d$, and therefore requires d boundary conditions. Usually the values of $\underline{U}$ on the boundary are given. From the general theory of elliptic systems [ADN] it follows that the linearized system (7.2) with such boundary conditions has one and only one solution, which has all the stability and smoothness properties one could expect. The theoretical results for the nonlinear system are more involved and the reader is referred to [T1].

7.2 Finite-difference approximations

The discretization is carried out on the same staggered grid as before (Fig. 6.1), using the difference equations

$$\sum_{i=1}^{d} \partial_i^h U_i^h = F_0^h \qquad \text{at cell centers} \tag{7.3a}$$

$$Q^h U_j^h + \partial_j^h P^h = F_j^h \qquad \text{at } j\text{-face centers} \quad (j=1,\ldots,d). \tag{7.3b}$$

where Q^h is some difference approximation to Q. Since $\det L^h = -\Delta^h (Q^h)^{d-1}$, it is clear that L^h is T-elliptic if and only if Q^h is T-elliptic, and L^h has good hT-ellipticity measure if and only if Q^h does. Hence, all we have to construct is a good approximation to Q. For small to moderate $hR|\underline{U}|$ (i.e., $hR|\underline{U}|$ not much larger than 1) this can be done by central differencing (Sec. 3.10.1). But for larger $hR|\underline{U}|$ upstream differencing (Sec. 3.10.2) or explicit artificial viscosity terms (Sec. 3.10.3) should be used. Either way, the resulting operator contains $O(h^p)$ artificial viscosity. A better multi-level possibility is to employ such Q^h only in the relaxation sweeps, while in the residual transfers use the central approximation (see Sec. 3.11).

7.3 DGS relaxation

Generalizing the scheme in Sec. 6.3 to *any* elliptic operator Q^h, relaxation proceeds as follows:

The j-th momentum equation (7.3b) is relaxed by changing values of u_j^h only, in a manner suitable for the operator Q^h. For example, if each component U_i has a constant sign throughout the domain, a point-wise Gauss-Seidel relaxation marching downstream is the most efficient manner: It gives a smoothing factor $\bar{\mu} \leq .5$, and $\bar{\mu} = O(hR|\underline{U}|)^{-1}$ for large $hR|\underline{U}|$. If all possible signs of U_i and all possible relative magnitudes of $hR|U_i|$ $(i = 1,\ldots,d)$ appear in the domain, symmetric line relaxation (for $d = 2$) or symmetric plane relaxation (for $d = 3$) is the best. Any line (or plane) direction may be chosen. Symmetric relaxation means that the sweep is made of two passes: In the first pass the lines are taken in some, say increasing, order, and in the second pass the order is reversed. In two dimensions, for

example, a double-pass relaxation sweep has a smoothing factor at most .2 for any (frozen) 5-point Q^h constructed by upstream differencing, hence its smoothing factor per one pass is $\bar{\mu} \leq .2^{\frac{1}{2}} = .447$.

Having relaxed in this way all the momentum equations ($j = 1,\ldots,d$) , we then make a pass of relaxation for the continuity equation (7.3a), by scanning the cells one by one in some consistent order. At each cell the relaxation step is a generalization of Fig. 6.2 and eqs. (6.14) above: Denoting again the center of the cell by $\underline{x}$ and its characteristic function by $\chi^h_{\underline{x}}$, the relaxation step is

$$u^h_j \leftarrow u^h_j - \delta h \partial^h_j \chi^h_{\underline{x}} \tag{7.4a}$$

$$p^h \leftarrow p^h + \delta h Q^h \chi^h_{\underline{x}} \quad , \tag{7.4b}$$

where δ is still given by (6.12) and (6.10). That is, δ is chosen so that the new velocities (7.4a) satisfy the discrete continuity equation at $\underline{x}$.

It is easy to see that changes 7.4 are such that the residuals of the momentum equations

$$r^h_j = F^h_j - Q^h u^h_j - \partial^h_j p^h \quad , \qquad (j = 1,\ldots,d) \tag{7.5}$$

are preserved, at least in the approximate sense of regarding Q^h as locally constant. Except for the omission of the lowest order term (see Sec. 7.1 and 7.4), this freezing of Q^h is in line with the usual assumption of the mode analysis (see Sec. 4). Hence it follows, as in Sec. 6.3, that the amplification matrix of a compound relaxation sweep has the form

$$R(\underline{\theta}) = \begin{pmatrix} \lambda(\underline{\theta})^N & \kappa_{01} & \cdots & \kappa_{0d} \\ 0 & \kappa(\underline{\theta})^K & & \bigcirc \\ \vdots & \bigcirc & \ddots & \\ 0 & & & \kappa(\underline{\theta})^K \end{pmatrix} \quad , \tag{7.5}$$

where $\lambda(\underline{\theta})$ is given by (6.15), N is the number of passes on the continuity equation included in the sweep, $\kappa(\theta)$ is the amplification factor (per pass) of the Q^h relaxation, and K is the number of passes on each momentum equation.

Usually one takes $N = K$, and the smoothing factor then is

$$\bar{\mu} = \max_{\frac{\pi}{2} \leq |\underline{\theta}| \leq \pi} \max(\lambda(\underline{\theta}), \kappa(\underline{\theta})) \quad , \tag{7.6}$$

so that we still have $\bar{\mu} = .5$ for two dimensional problems with symmetric line relaxation (in the momentum equations). In case the flow in the entire field is in the same general direction, one can use downstream relaxation (for the momentum equations) with $N > K$, since the momentum-equations smoothing is faster than the continuity-equation one (cf. Sec. 2.3).

7.4 Multi-grid procedures

The grids, their relative positions and the interpolation procedures between them are the same as for Stokes equations (Sec. 6.4). Because of the nonlinearity, FAS should of course be used, and the full weighting (6.19) is preferable to (6.18) in the fine-to-coarse transfers of both the solution and the residual function.

For large values of R , the effect of physical instability (see Sec. 7.6) is felt as deterioration in the smoothing and convergence rates of relaxation on the coarse grids. (This can be regarded as the effect of the lowest-order term, which was neglected in the smoothing analysis). Hence it is necessary at large Reynolds numbers to work with accommodative algorithms (Sec. 2.2). Such algorithms spend more sweeps and cycles at coarse levels. The overall efficiency, however, is not too much affected, since those extra sweeps and cycles cost very little.

7.5 Numerical results

Our codes are still in a stage of development: We programmed only two-dimensional problems in rectangular domains. More importantly, the symmetric line relaxation for the momentum equations is not yet implemented. Only pointwise Gauss-Seidel was so far used. Hence the performance is not optimal for problems with reverse flows and large Reynolds numbers. On the other hand we could fully check the efficiency of our procedure, even for large R , for problems where our relaxation marching direction is everywhere downstream. For such problems the numerical results really fulfill all the

theoretical expectations.

We first experimented with cycling algorithms. For small and moderate Reynolds numbers the behavior was essentially as for Stokes equations, namely the convergence factor per relaxation-work-unit was about .50 to .57 for the first few cycles, and increased asymptotically to around .65. For large Reynolds numbers $(50 < R < 4000)$ the convergence was even faster (sometimes much faster) in the first cycle, but it slowed down later. The asymptotic rate is not much worse than .65 as long as R does not approach physical instabilities. (In the example shown in [D1] the asymptotic convergence factor for $R = 4096$ seems to be around .78 on a 64×64 grid. But the procedure there is not optimal since it contains too much work per cycle.) A collection of results of cycling Navier-Stokes algorithms can be found in [D2].

Full Multi-Grid (FMG) algorithms were then constructed and we studied the main question of multi-grid performance: How much work is required to solve the difference equations "to within the truncation-errors", i.e., to the point where the numerical-solution errors $\|u^h - U\|$, $\|v^h - V\|$, $\|p^h - P\|$ are comparable to the discretization errors $\|U^h - U\|$, $\|V^h - V\|$, $\|P^h - P\|$, where (P,U,V) denotes here the trace of the true differential solution on the finest grid, whose meshsize is h , and where the norm is the maximum norm. Since the true solution is seldom known, we took for our tests either one of the following approaches: (i) Specifying (P,V,U) in advance, we computed from them both the forcing terms (F_0,F_1,F_2) , to serve in our equations, and the boundary conditions. Such problems turn out to be somewhat artificial. (ii) Instead of comparing with (P,U,V) , we compared with $(P^{h/2},U^{h/2},V^{h/2})$, the solution on a still finer grid, obtained there by many multi-grid cycles.

Our tests confirmed the theoretical prediction that, since the smoothing rate is .5, the FMG algorithm with only one multi-grid correction cycle (similar to Fig. 2.2, but possibly with more cycling on coarse levels) always produces a solution within the truncation errors. The work of such

algorithm was always less than 10 relaxation-work-units. With a more precise choice of the algorithm parameters, we could get the work down to 5 - 7 work units. The results are fully reported in [D2]. Here we only reproduce a small example - Table 7.1.

The solution errors for two problems are shown. The results are grouped in pairs: the lower entry in each pair is the error for our 1-cycle algorithm, while the upper entry is the result of many-cycles algorithm, effectively giving the discretization error. The domain of the two problems is the unit square. The boundary conditions for problem I are

$$U = \begin{cases} 0 & \text{top and bottom boundaries} \\ 7y(1-y)^2 & \text{left boundary} \\ 7y^2(1-Y) & \text{right boundary} \end{cases}$$

and $V = 0$ on all boundaries, and for problem II

$$U = \begin{cases} 60y^2(1-y)^2(2y-1) & \text{right boundary} \\ 0 & \text{other boundaries} \end{cases}$$

and $V = 0$ on all boundaries. Our 1-cycle algorithm contains at the end two relaxation sweeps on the finest grid. At most one of them is really needed in order to obtain the level of errors shown in the table. Hence the shown amount of work units could be reduced by at least 1. Note that Problem II contains a backward flow, and that only forward relaxation was used. With symmetric relaxation the results for $R = 100$ should be improved. Note also that we used weighted averages of central and upstream differencing. Hence for $Rh \gg 1$ our accuracy is $O(h)$.

TABLE 7.1

Problem	R	Grid h	$\frac{\|U^h - U^0\|}{\|u^h - U^0\|}$	$\frac{\|V^h - V^0\|}{\|v^h - V^0\|}$	$\frac{\|p^h - p^0\|}{\|p^h - p^0\|}$	Relax. Work Units
I	8	8 × 8	.00580 .00577	.00680 .00728	.0660 .0623	23.3 9.6
		16×16	.00147 .00157	.00170 .00181	.0159 .0171	26.4 7.7
		32×32	.000376 .000417	.000417 .000438	.00385 .00475	26.8 7.2
	100	16×16	.00380 .00376	.00383 .00372	.374 .331	28.9 9.2
		32×32	.00153 .00154	.00237 .00240	.114 .115	27.3 7.9
	500	32×32	.00379 .00407	.00288 .00367	1.730 1.828	43.4 9.7
II	0	32×32	.000476 .000498	.000661 .000748	.00956 .01240	27.2 7.1
	8	16×16	.00178 .00196	.00281 .00289	.0643 .0830	26.8 7.7
		32×32	.000502 .000589	.000679 .000776	.0156 .0221	28.4 7.3
	100	16×16	.0292 .0384	.0315 .0402	2.478 3.457	47.7 10.4
		32×32	.0149 .0207	.0170 .0234	1.236 1.831	45.1 9.8

(p^0, u^0, v^0) is the solution on the 64×64 grid.

7.6 Physical instabilities

The main difference between physical and numerical instabilities is that the latter first appear at high-frequencies (where the numerical solution does not approximate the differential one) while the former first appear at low-frequency modes, whose Reynolds number (i.e., R times their wavelength) is large. The slow divergence of such

smooth modes in a relaxation process (which may be regarded as a time-like process) does not trouble the error-smoothing process. Also, the instability does not appear when the multi-grid process (see Fig. 2.2) first works at coarse levels, since the numerical scheme contains enough artificial viscosity (see Sec. 3.10). It is only when the process comes back to coarse levels after visiting sufficiently fine levels (where artificial viscosity is sufficiently small) that the physical instability starts to show up.

It is clear that no purely iterative (time-like) solution process can solve the steady-state flow equations when the solution is unstable. In the multi-grid process, however, this limitation is, in principle, removed, since, at each cycle, the coarsest-grid equations can be solved directly, not by relaxation. *It is in this way possible to calculate unstable solutions*, provided the coarsest grid used is fine enough to resolve the unstable modes.

REFERENCES

[ADN] Agmon, S., A. Douglis, and L. Nirenberg, Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions II, Comm. Pure Appl. Math. 17, pp. 35-92, 1964.

[B1] Brandt, A., Multi-Level Adaptive Technique (MLAT) for Fast Numerical Solution to Boundary Value Problems. Proceedings of the 3rd International Conference on Numerical Methods in Fluid Mechanis (Paris, 1972), Lecture Notes in Physics 18, pp. 82-79, Springer-Verlag, Berlin and New York, 1973.

[B2] Brandt, A., Multi-Level Adaptive Techniques, IBM Research Report RC-6026, IBM T.J. Watson Research Center, Yorktown Heights, N.Y., 1976.

[B3] Brandt, A., Multi-Level Adaptive Solutions to Boundary-Value Problems, Mathematics of Computation, 31, pp. 333-390, 1977.

[B4] Brandt, A., Multi-Level Adaptive Solutions to Partial-Differential Equations – Ideas and Software, Proceedings of Symposium on Mathematical Software (Mathematics Research Center, University of Wisconsin, March 1977), (John Rice, ed.), pp. 277-318, Academic Press, New York, 1977.

[B5] Brandt, A., Multi-Level Adaptive Techniques (MLAT) and Singular-Perturbation Problems, Proceedings of Conference on Numerical Solution of Singular Perturbation Problems (University of Nijmegen, The Netherlands, May-June 1978). Also appears as ICASE Report 78-18.

[B6] Brandt, A. Lecture Notes of the ICASE Workshop on Multi-Grid Methods. With Contributions also by J.C. South (Lecture 8), J. Oliger (10), F. Gustavson (13), C.E. Grosch (14), D.J. Jones (15), and T.C. Poling (16). ICASE, NASA Langley Research Center, Hampton, Virginia, 1978.

[B7] Brandt, A., Multi-Level Adaptive Computations in Fluid Dynamics, Proc. of the AIAA 4th Computational Fluid Dynamics Conference, Williamsburgh, Virginia, July 1979.

[BDR] Brandt, A., J.E. Dendi, Jr., and H. Ruppel, The Multi-Grid Method for Semi-Implicit Hydrodynamics Codes, LA-UR 78-3066 report of Los Alamos Scientific Laboratory, Los Alamos, New Mexico, 1978.

[BH] Bramble, J.H. and B.E. Hubbard, On the formulation of finite difference analogues of the Dirichlet problem for Poisson's equation, Numerische Mathematik, 4, pp. 313-327, 1962.

[C1] Ciarlet, P.G., The Finite Element Method for Elliptic Problems. North-Holland Publishing Co., New York, 1978.

[D1] Dinar, N., On several aspects and applications of the multi-grid method for solving partial differential equations, NASA Contractor Report 158947, NASA Langley Research Center, Hampton, Virginia, 1978.

[D2] Dinar, N., Fast methods for the numerical solution of boundary-value problems, Ph.D. Thesis, Weizmann Institute of Science, Rehovot, Israel, 1979.

[DN] Douglis, A., and L. Nirenberg, Interior estimates for elliptic systems of partial differential equations, Comm. Pure Appl. Math., 8, pp.503-538, 1955.

[F1] Frank, L., Coercive B.V.P. with small parameter, C.R. Acad. Sci., Paris, t.282, Ser. A., pp.1109-1111, 1976.

[F2] Frank, L., Difference operators and singular perturbations, C.R. Acad. Sci., Paris, t.283, Ser. A., pp. 859-862, 1976.

[F3] Frank, L., General boundary value problems for ODE with small parameter, Annali di Mat. Pura ed Applicata, (IV), Vol. CXIV, pp. 27-67, 1977.

[F4] Frank, L., Coercive singular perturbations: Stability and Convergence, Proc. of the Conf. on the Num. Anal. of Sing. Pert. Problems, Nijmegen, Holland, 1978.

[F5] Frank, L., Coercive singular perturbations I: A priori estimates, Annali di Mat. Pura ed Appl., (to appear).

[F6] Frank, L., Difference singular perturbations, I: A priori estimates, (to appear).

[LW] Lax, P.D., and B. Wendroff, Systems of conservation laws, Comm. Pure Appl. Math., 8, pp. 217-237, 1960.

[MT] MUGTAPE. A Tape of Multi-Grid Software and Programs. Distributed at the ICASE Workshop on Multi-Grid Methods. Contributions by A. Brandt, N. Dinar, F. Gustavson, and D. Ophir, 1978.

[P1] Poling, T.C., Numerical experiments with multi-grid methods, M.A. Thesis, Department of Mathematics, The College of William and Mary, Williamsburg, Virginia, 1978.

[O1] Ophir, D., Language for processes of numerical solutions to differential equations, Ph.D. Thesis, Mathematics Department, The Weizmann Institute of Science, Rehovot, Israel, 1979.

[S1] Shiftan, Y., Multi-grid method for solving elliptic difference equations, M.Sc. Thesis (in Hebrew), The Weizmann Institute of Science, Rehovot, Israel, 1972.

[SB] South, J.C., Jr., and A. Brandt, Application of a multi-level grid method to transonic flow calculations, ICASE Report 76-8, NASA Langley Research Center, Hampton, Virginia, 1976.

[T1] Temam, R., Navier-Stokes Equations, North-Holland Publishing Co., Amsterdam, New York, Oxford, 1977.

[T2] Thomée, V., Elliptic difference operators and Dirichlet problem. Contributions to Differential Equations, 3, pp. 301-324, 1964.

[T3] Thomée, V., Discrete interior Schauder estimates for elliptic difference operators, SIAM J. Numer. Anal. 5, 1968.

[T4] Thomée, V., On the convergence of difference quotients in elliptic problems. Numerical Solutions of Field Problems in Continuum Mechanics, SIAM-AMS Proc., Vol. II, pp. 186-200, Am. Math. Soc., Providence, R.I., 1970.

[T5] Thomée, V., Convergence near plane boundaries of some elliptic difference schemes, Numer. Math., 17, pp. 45-53, 1971.

[TW] Thomée, V., and B. Westergren, Elliptic difference equations and interior regularity. Numerische Mathematik 11, pp. 196-210, 1968.